• THE ULTIMATE TABLEBOT • ROBOTS WHO LIVE WITH US •

# SERVO MAGAZINE

FOR THE ROBOT EXPERIMENTER

*www.servomagazine.com*

*April 2006*

# MEET eMo

***Get a Feel(ing) for What Lies Ahead in Robotic Companionship***

- Hacking the E-Maxx
- Closed Loop Systems
- Anatomy of a Servo

# SERVO

## Features & Projects



## ON THE COVER

**eMo and his creator Professor Noel Sharkey**



*SERVO Magazine* **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. APPLICATION TO MAIL AT PERIODICALS POSTAGE RATE IS PENDING AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

04.2006
VOL. 4 NO. 4

## Departments



## Columns

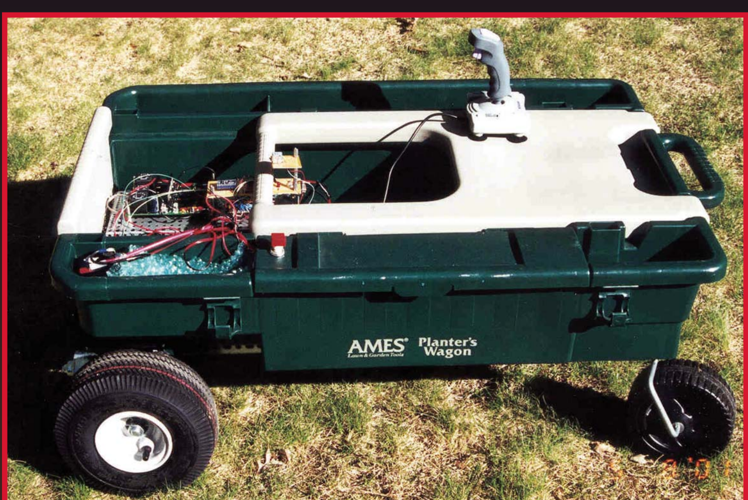


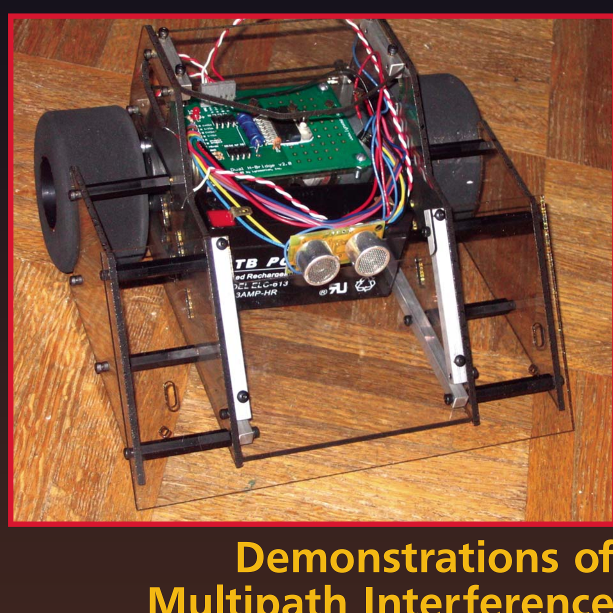
### SPECIAL BYE-BYE TO AIBO



# Coming 05.2006

### Build an Advanced Two DC Motor Test Platform



### Demonstrations of Multipath Interference

# Mind / Iron

by Dave Calkins

## R.I.P. Aibo

*Or not.*

Let us now, dear mourners, look back on the last five years of the finest consumer robot produced to date. Sony has discontinued its Aibo division. After producing 150,000 units, the end is now in sight.

But before we plan the wake, let's be realistic. I have several Aibos myself, but Sony didn't break down my door at 3 am and take them away. They just stopped making new ones. They didn't kill the dog, they simply had it spayed so it wouldn't have any puppies. While I was dismayed when I first heard the announcement, I have come to agree with it. Not because I don't like Aibos, but because this move will help advance robotics.

When beloved products go away, are they really gone? You can always try eBay. If a TV show is discontinued, fans might start a letter campaign to the network, but it's probably not coming back. Why not? Because it's run its course. There are those fans who want time to stop and never go forward. Why? Because of the sheer joie de vie of the moment. Like a first kiss, a cherry high, or a new toy (which is what Aibos were), we want to stay in that second in time, to hold on to the innocence and newness of it. But roomy new castles are built on the ruins of small old forts, and continuing to add-on to an extant product ensures mediocrity, not excellence.

VHS could have been replaced by a similar but improved S-VHS, but instead we moved on to laserdiscs, which gave way to DVDs, which are even now giving way to DVRs and video-on-demand. These are technologies we never imagined 20 years ago. Can't you believe that there is something better than foot-long robot dogs in our near future? Stopping production of the Aibo ensures one thing: Something better is coming down the pipe from some garage inventor or hungry company. If a need truly exists, someone will find a way to fill it.

And let us not forget that the Aibo was not without problems. Three major faults come immediately to mind.

**Proprietary OS.** Oh, they called it open source, but the last time I checked, Panasonic and AiboPet weren't allowed to sell or advertise Aibo add-ons. Memories of Beta, anyone? There's a reason everyone ended up with VHS, and it wasn't superior quality. It was JVC letting the format be open for other companies to license and build upon. Sony never let Aibo go to the park and mate with other doggies. Cross-breeding is as good for plastic dogs as it for furry ones.

**Expandability.** Imagine a computer without a hard drive. You boot from your CD. You want to use PhotoShop? Boot. Excel? You must re-boot. Want to run both at once? Sorry, you're out of luck. Aibos needed a 500 meg or so on-board hard drive with the OS and personalities stored there. If I want to run Navigator, I shouldn't have to remove all the memory and install a different set, and then re-boot — losing my dog's personality. I should

**Mind/Iron Continued** →

Published Monthly By
**T & L Publications, Inc.**
430 Princeland Court
Corona, CA 92879-1300
**(951) 371-8497**
FAX **(951) 371-3052**
Product Order Line **1-800-783-4624**
**www.servomagazine.com**

Subscriptions
Inside US **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

**PUBLISHER**
Larry Lemieux
**publisher@servomagazine.com**

**ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING**
Robin Lemieux
**display@servomagazine.com**

**CIRCULATION DIRECTOR**
Tracy Kerley
**subscribe@servomagazine.com**

**WEB CONTENT/STORE**
Michael Kaudze
**sales@servomagazine.com**

**PRODUCTION/GRAPHICS**
Shannon Lemieux
Michele Durant

# BIO-FEEDBACK

*Dear SERVO:*

I received my January 06 issue of *SERVO* and jumped right into Gerard Fonte's "Programmable Logic — Part 1" article. The article was very good and it's just the type of article I want to see more of.

However I have some questions — on Page 30, Figure 10, does the truth table and the reduced truth table match the schematic? In the truth table, should the "Input 1 Comparison" column read "inverted" for the inputs of 6 and 7 instead of "follows"? Also in the reduced truth table should the "OUTPUT" column read "inverted input 1" for the input of 3 instead of "follows input 1"? Also, in order to have the circuit depicted in Figure 9 equivalent to the circuit in figure 10, should D3 of Figure 10 be tied to "Not Input 1" instead of "Input 1"?

**David Ellis**
**Arlington, TX**

*Writer response:*

*Mr. Ellis is correct on all counts. I screwed up. For whatever reason, I managed to equate "0,1" with "1,0" in the main truth table and then propagated that error into the schematic and reduced truth table. A corrected Figure 10*

*Continued on Page 77*

be able to just add new tricks to the extant personality. I can teach my flesh-and-blood dog to both sit and fetch. I don't have to swap her brain every time I want a new trick.

**Marketing.** Few people outside of the robot community really knew of Aibo's full capabilities. Its wireless video, Internet controllability, emerging personas, trainability, facial recognition, and many other features were unknown to the average consumer. They saw it as a $2,000 toy, and most people rejected it. If Sony had done a better job of educating people about everything that Aibos could do, sales would have increased ten-fold. Consumers just didn't know how advanced Aibos really were.

So to some extent, the Aibo was doomed from the get-go. Like so many of Sony's product lines, Aibos were crippled not because they weren't brilliant in concept, but because Sony is so foolish about licensing and marketing. And Sony has often shown that they are incapable of learning from their previous mistakes.

As to the Aibos themselves? They were a brilliant beginning. A taste of the future. A starting point. But they were mudskippers on the evolutionary tree, and it's time to lose the rhetorical gills and develop fingers. The list-serve for RoboCup soccer teams who reprogram Aibos to play soccer is now abuzz with pleas to Sony to keep the Aibo. But instead, I would argue that all of these brilliant roboticists should move forward! Should we all still be programming on 386s? Instead of upgrading Windows yet again, why not switch to Linux? And if Linux doesn't exist, write it.

Let's build a better walking, camera-capable, open-source platform. Just as jet engines replaced propellers on airplanes, so must some other robot replace Aibos. The point isn't for us to be stuck on modifying a single platform, but to innovate! To create! To let this sour grape force us to plant a much sweeter fruit.

I still have my old 486 laptop with Windows 3.1 — I turn it on about once a year to reminisce about what once was and remind myself how much better things are now. I rather expect that in just five years, I'll bring out my dusty Aibos and marvel at how limited they were compared to my new robots. There was robot life before the Aibo, and it will continue after the Aibo. I'll forever love the Aibo, but I see it for what it was — a sweet, old-fashioned king who must be replaced by a young, forward-thinking prince.

The Aibo is dead! Long live the Aibo! **SV**

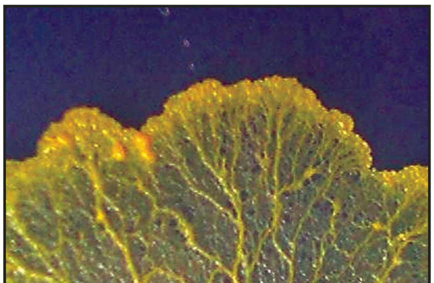
# Robytes

by Jeff Eckert

**Are you an avid Internet surfer who came across something cool that we all need to see? Are you on an interesting R&D group and want to share what you're developing? Then send me an email! To submit related press releases and news items, please visit www.jkeckert.com**

***— Jeff Eckert***

## Slime-Directed Hexapod



***A portion of Physarum polycephalum used to control the movement of a hexapod robot using the cellular robot controller shown in the diagram below. Photo courtesy of the University of Southampton.***

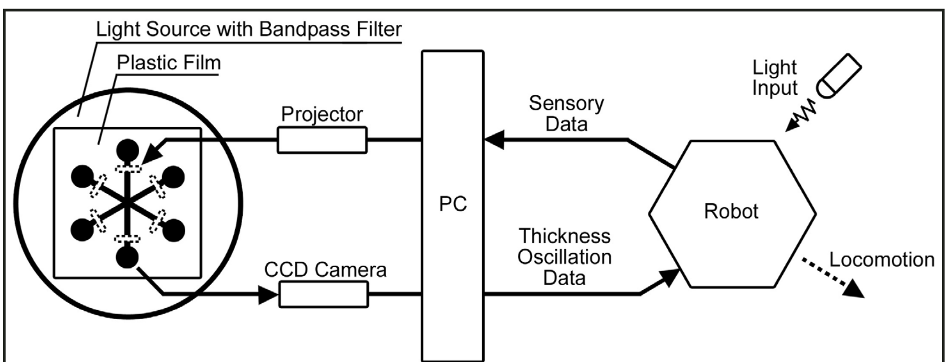
If you are a serious gardener, it is quite possible that you have noticed a slimy yellow blob growing in your mulched flower bed. This stuff is known scientifically as *Physarum polycephalum*, but is described by the Mold Help organization (**www.mold-help.org**) as looking "like dog vomit."

Technically, it is an amoeboid plasmodium, which basically means that it's a bunch of protoplasm that has the ability to move around autonomously like an amoeba via rhythmic contractions within the organism. It's not a highly desirable substance unless you are working to create lifelike adaptive behavior in a very simple robot, in which case it appears to come in handy.

A paper published by some scientists from Kobe University (Japan) and the University of Southampton (UK) describes a concept by which movement of the slime mold — which tends to avoid light — can be used to make a hexapod robot behave in the same manner. Basically, they designed a plastic mask that allows the slime to be configured as six oscillators in a star pattern, with one corresponding to each leg of the bot. When light is projected on one of the oscillators, it responds, and the response is detected by one of six light sensors that, via six channels of a PC sound card, control the six servos of the legs.

Although the experiments are in an early stage, the authors "expect robust biological cells such as those of molds and thermophilic bacteria to become an integral part of technological devices." They further believe that the concept may have applications in nanofabrication as a way of "obtaining highly integrated, robust information processors, at least for niche applications."

As of this writing, you can download the complete paper by aiming your browser at **http://eprints.ecs.soton.ac.uk/11749/01/TsudaS06RobotCircToCell.pdf** (note that the document name is case sensitive).




***Cellular robot controller diagram. Photo courtesy of the University of Southampton.***

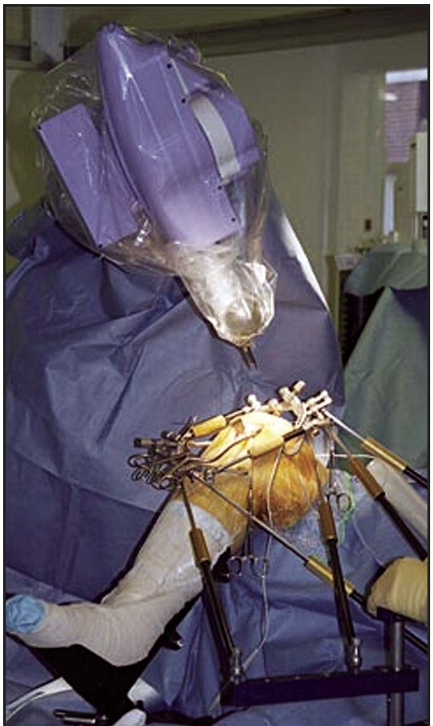
## Rescue Bot Introduced

***Tmsuk's Enryu, adapted for snow removal and rescue. Photo courtesy of Tmsuk Co.***

On a larger and probably more practical scale is Enryu, which is Japanese for "rescue dragon." For the English-speaking world, it is subnamed the T-52 "Hyper Rescue Robot." Originally developed by Tmsuk Co. (**www.tmsuk.co.jp**) for earthquake rescue operations, it has now been adapted for use in snow. In a recent test conducted in the heart of Japan's snow country, it successfully retrieved a snowbound car.

The unit stands 11.5 ft tall, weighs five tons, and moves at three km/hr on tank treads. Its two 15-ft hydraulic arms can each lift up to 1,100 lbs. Enryu is also fitted with seven cameras, allowing it to be remotely operated in hazardous areas, although it does have a control chamber with enough room for one operator. There seem to be no current plans to employ Enryu in the US, but if you happen to be skiing in Japan's Niigata prefecture, don't be surprised if one taps you on the shoulder and hands you a warm slug of sake.

## Bot Improves Knee Surgery



Acrobot — a robotic surgical assistant — helps with knee replacement. Photo courtesy of Imperial College London.

Back in February, a team of surgeons and engineers at the Imperial College London (**www.imperial.ac.uk**) conducted tests on 27 patients undergoing partial knee replacement surgery to determine whether the Acrobot surgical assistant could improve patient outcomes. The patients were separated into two groups as part of a randomized controlled trial, with 14 having conventional surgery and the remaining 13 having robot-assisted surgery.

Although the operations took a few minutes longer using the robotic assistant, the replacement knee parts were more accurately lined up than in conventional surgery. It turned out that all of the robotically assisted operations lined up the bones to within two degrees of the planned position, but only 40 percent of the conventionally performed cases achieved this level of accuracy.

It probably should be noted that the builder of the bot, Acrobot Co. Ltd. (**www.acrobot.co.uk**), is a spin-off of the Imperial College, but it still looks like your local hospital could put the device to good use.

## R.I.P. Aibo?

Sony's Aibo robotic dog will bark no more. Photo courtesy of Sony Corp.

Finally, we must sadly note that, as part of a program of cost-cutting and reorganization, Sony (**www.sony.com**) has decided to put Aibo — the robot dog — to sleep. Although the pooch's website (**www.sony.net/Products/aibo/index.html**) was still operating as of this writing, it has been widely reported that his demise is imminent. The company has sold more than 150,000 of the $2,000 canines, but it appears that Aibo can no longer compete with lower-priced mongrels. Farewell, old buddy.

## Unmanned Defense Vehicle

Even more monstrous is the unmanned palletized load system (PLS) recently demonstrated by Oshkosh Truck Corp. (**www.oshkoshtruckcorporation.com**), designed to allow convoy resupply missions to be carried out without the need to put live soldiers in harm's way. The unit has been under test for two years at the DARPA Grand Challenge races and has undergone further testing in desert environments similar to what exists in the Middle East. The manned version has already been operated in places like Bosnia, Kosovo, Afghanistan, and Iraq.

A 10-wheel-drive truck and trailer system, the PLS is designed to transport containers carrying ammunition and other critical supplies or large tanks that contain fuel or water. The machines have a 16.5-ton payload capacity and an on-board material handling system that quickly unloads and loads cargo. **SV**

Oshkosh's unmanned palletized load system, designed for military use. Photo courtesy of Oshkosh Truck Corp.

www.robobusiness2006.com

# ROBO Business 2006

## CONFERENCE & EXPO

**June 20-21, 2006**
**Sheraton Station Square Pittsburgh, PA**

**The International Business Development Event for the Mobile Robotics and Intelligent Systems Industry**

### Five Comprehensive Tracks

- Business Development and Partnership
- Technology and Standards
- Applications and Products
- Investment Opportunities
- Markets and Industries
  - Security and Defense Robotics
  - Intelligent Transportation and Field Robotics
  - Healthcare Robotics
  - Consumer Robotics


Photos courtesy of iRobot, Mobile Robotics and Openware Robotics.

Founding Sponsor: iRobot

Premier Sponsor: Innova Holdings, Inc. Bridging Innovative Technologies; The Technology Collaborative Accelerating Digital & Robotic Innovation.

Gold Sponsor: CoroWare

Silver Sponsors: Advanced Digital Logic Inc; Applied Perception; Autonomous Solutions Inc.; Automatika; Evolution Robotics; Foster-Miller A QinetiQ Company; JETRO Japan External Trade Organization; Robot Technology Osaka; re2; Segway; Skeyes Unlimited Corporation; Valde Systems, Inc.; Yost Engineering, Inc.

Association Co-sponsors: GETI Global Emerging Technology Institute; Mass Technology Leadership Council; MDG boston forum; OMG Object Management Group; Robot Hall of Fame powered by Carnegie Mellon; Robotics Society of America

Produced by: roboticstrends; IDG World Expo

Premier Media Sponsor: Popular Science

Academic Co-sponsor: Carnegie Mellon

# GEER HEAD

by David Geer

Contact the author at geercom@alltel.net

# eMo

## A Robotic Expert in Non-Verbal (And Even Some Verbal) Communication

### "Feelings, oh, oh, oh, oh, oh ..." You Get the Idea.

I am reminded of a famous song lyric, as I often am when writing about robots — songs and robots, strange bedfellows indeed! Although I had thought of referencing Barry Manilow's "Feelings" (eMo, oh, oh, oh, oh, oh, eMo), it would have been more humorous than illustrative.

Actually, the first song that came to mind was "Frosty the Snowman" — "with a corn cob pipe and a button nose and two eyes made out of coal!"

Frosty's static face came to life along with the rest of him, as the story goes. The impetus for eMo the emotive robot head, which we will be discussing (and, perhaps, for the tale of Frosty, too), is the personification of lifeless things for our benefit, so we can relate to them.

### The AI Bot Cometh

In the 21st century, we are seeing more and more so-called intelligent machines becoming a part of our lives, both at work and at home (Roomba the robot vacuum cleaner, Industrial robots, etc.).

As these domestic companions and co-workers become decidedly more adept, it is important that they behave in a way that is meaningful in a world of human communicators, according to Professor Noel Sharkey, creator of eMo.

"For us, expressions and body language are as important as what is

#### ROBOTIC GROWING PAINS

eMo is a stepping-stone toward greater achievements in the world of robotic emotional expression. Professor Noel Sharkey and team are currently working on a new robot named the Creative Interaction Machine (CIM).

CIM is expected to lead to the evolution of the emotive powers of eMo. "Now that we understand much more about the problems around developing an interactive head, we can greatly improve on it," says Professor Sharkey.

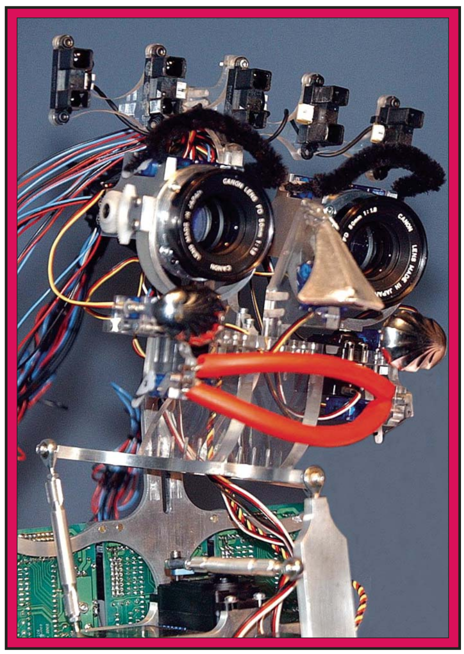
The CIM team consists of three lecturers of varying backgrounds in computer hardware, gaming, and machine learning, including a professor of speech. Their hope is to develop a more interactive humanoid robot that can converse and perhaps even hear.

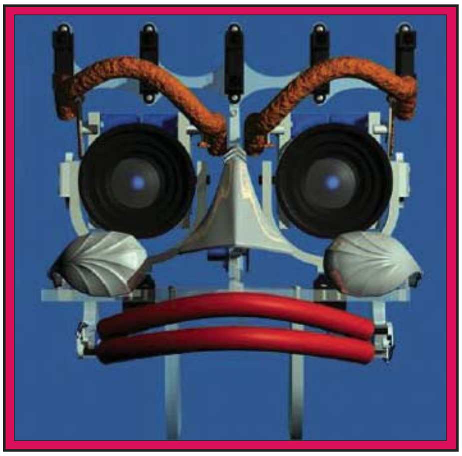
The first step toward a better emotive robot was to replace the old computer system with a miniature motherboard. This should help CIM become more portable and make it possible to someday give him a body, as well.

Color tracking cameras have been added in the CIM version so that the creators can have much greater breadth of control over head movements in response to tracking objects and people.

Speech computer chips to enable the gift of gab for CIM are being tested. Once the funding is raised, construction of a body for CIM should follow, which will house and employ many more sonar sensors incorporated directly into the body.

This is a shot of eMo alone.





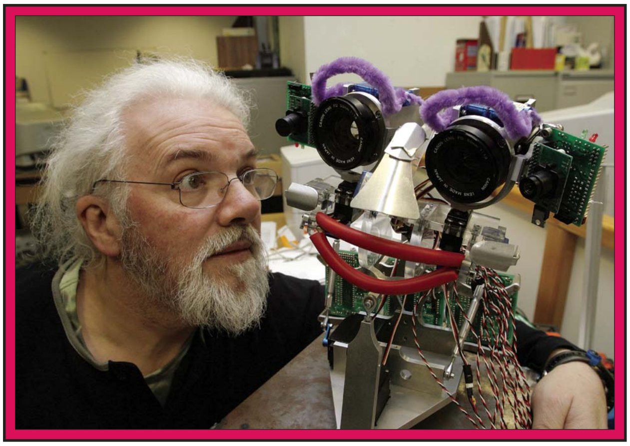
eMo expressing anger.

eMo's lips all by themselves.

eMo expressing joy, happiness with the aid of his servos and computer brain (the computer is separate, not attached).

actually said; sometimes more so. This is why we need to study mechanical expressions and body movement (that shrug of the shoulder) so that we can understand our machines in a way that suits us," says Professor Sharkey.

Here, we see Professor Noel Sharkey looking at CIM, the next generation of eMo the emotive robot head.



That's the long-term intention of today's simpler emotive robots — to know that, when someday robots and people are working side-by-side, whatever the job, they will be able to reflect our emotions or pretend their own.

It's expected that these capabilities will be needed in our soon-to-be working companions because of our own need to relate emotively to robots that will work and act so much more like us by that time.

Why do you think we appreciate our pets so much? How many of you that have pets either genuinely see a smile or sadness on the face of your pet, or imagine one so that you can relate to it? If we didn't perceive pet emotions, how would they give us comfort? I'm not saying pets don't really have emotions, just that the emotive aspect of something living is crucial to our relating to it.

So, here we are a long way out from artificially intelligent robots in human likeness. We're investigating the idea of how to make robot faces that can form believable human expressions with a minimum of technical resources.

"This is why there is no face or skin on eMo," says Professor Sharkey. But, even with all the technical elements bare to the world, eMo's simulated emotions shine through.

### LOOK OUT FOR THAT ... CONCRETE BENCH!

eMo's first show opened with an ouch! That day in Birmingham (UK) at the ThinkTank scientific museum, Professor Noel Sharkey was there with his team, making last minute adjustments on the eMo system and waiting for the arrival of the news media.

This was an intense situation as everything had to be perfect and the professor had to take time to spend with the members of the press. He also had to leave for China early the next day to run their National Creative Robotics Competition.

When the media began to shuffle in, Professor Sharkey walked over to greet the first TV crew, hand outstretched to meet them. "Instead of finding a handshake, I found myself at the camera woman's feet writhing in agony. Not really the best introduction," says Professor Sharkey.

You see, the room went dark at that precise moment and the professor ran into a concrete bench in his path. A real trooper, the professor gave an interview before being rushed off for medical care.

## Facial Elements

eMo's face consists, first, of an aluminum nose with a ridge at the top that can be lifted up and down to help form an expression of disgust. After all, it would be neither humanoid nor humane to not instill both positive and negative — and, yes, even righteous — emotions

in eMo.

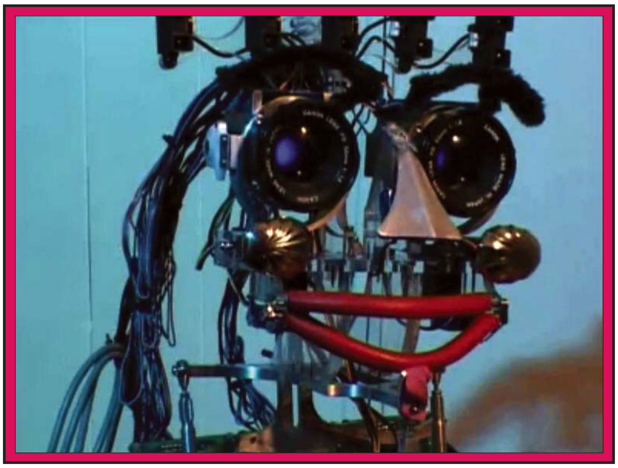
It would be awkward for eMo to be in a situation where disgust would be the appropriate response and, yet, have him be unable to give such response.

eMo's eyes are camera lenses. This was a simple, yet, perfect solution as the iris of the lens can be opened or closed to appear like widening or narrowing human pupils, which also contribute to showing emotion with the face.

Our pupils tend to enlarge when we are attracted to someone and grow small when we dislike them, according to Professor Sharkey.

eMo's eyebrows are made out of pipe cleaners, which move up and down to help him express anger or surprise. Though not a cheeky fellow, eMo has moving cheeks formed from two spoons, which simulate the part that real moving cheeks play when forming a smile.

eMo's mouth is made of sturdy wire surrounded by a red silicon tube. His neck tilts back when expressing surprise and to the side "for a cutesy effect," says Sharkey.



Angle view of eMo in happy mode, showing more of his technology and wiring.

Another shot of Professor Sharkey with eMo.

## Keeping eMo's Emotions in Check

The motion of eMo's emotive physical features is orchestrated individually by one or more servos; the mouth employs two servo motors. Most of these are the kinds of servos you find in model racing cars and vary in size, depending on the load they must bear. The neck requires a large industrial servo, normally used for the winching of sails on a vessel.

Twenty-one servos in all are connected directly to a computer. Changing eMo's facial expression means sending 21 "numbers" to and from the computer and the servos to tell each servo the degree to which they must turn. This is controlled automatically by a computer program written in C.

## So, How Do You Feel, eMo?

eMo shares the five most common human expressions: joy, sadness, anger, surprise, and disgust. A sixth is a neutral expression, if you count it.

eMo is on permanent display at the ThinkTank in Birmingham, UK. There, he is enabled by five infrared distance sensors on the crown on his head (King eMo?), which help him interact with passers' by. Here's how eMo responds, thanks to these sensors.

When someone moves in close, eMo pulls back. eMo looks at people when he detects them with his sensors. "The sensors send a signal about the location of people to the computer and it sends servo values to eMo so that the robot points at the person," says Sharkey.

For accuracy, because of parents carrying children and children running back and forth, the level of sensitivity of his sensing and corresponding movements is greatly limited. Otherwise, he would be constantly bopping all over the place.

A modestly priced home security sensor was fitted to the wall of the ThinkTank gallery, which was also connected to eMo's computer brain. With this, eMo automatically responds to people who are just entering the gallery by turning, looking at first surprised and then happy, winking at his new guests.

Finally, while gallery visitors are watching eMo, he will go into his "chat mode" in which he moves his head around animatedly, opening and closing his mouth emitting a "high-pitched babbling sound" (what a chatter box, that eMo!).

## eMo is Not Alone

The world of emotionally expressive robots is by no means small. Other examples include the work of Fumio Hara of the University of Science, Tokyo, who created a humanoid female face with silicone skin. Hara used

### THE THINKTANK

The ThinkTank — Birmingham's science museum — is located at:

Millennium Point, Curzon Street, Birmingham, B4 7XG, West Midlands, England. The phone number is 0121 202 2222. The hours of operation are 10 am to 5 pm with final admission at 4 pm, seven days a week. The ThinkTank is closed on Christmas Eve, Christmas Day, Boxing Day, and the 27th of December.

### RESOURCES

Links to eMo, with links to video of eMo and further information (navigation is a little tricky)
**www.dcs.shef.ac.uk/~noel/Media/dynamic2.htm**

The ThinkTank where eMo resides
**www.thinktank.ac/**

Links to research and discourse on emotional robots
**www.bartneck.de/work/bartneck_robofesta.pdf**

**http://fcis.oise.utoronto.ca/~aviseu/eng_emorobots_content.html**

**www.mindjack.com/feature/**

**emachines.html**

**http://web.mala.bc.ca/clemotteo/Pandora/Phil%20362/should_robots_feel.htm**

**www.corante.com/brainwaves/archives/2005/08/26/emotional_robots.php**

**www.elearning-reviews.org/topics/human-computer-interaction/usability/2004-norman-emotional-design/**

**www.aaai.org/AITopics/html/emotion.html**

**www.emotionalmachines.com/**

hydraulics to make lifelike muscle movements in the otherwise lifeless feminine face.

In another example, a furry little robot seal named Paro is seeing use in therapy for the very young and very old in nursing homes. There is also Feelix, the creation of Jakob Fredslund and Dolores Camero of the Aarhaus University, Denmark. This feeling bot was made from LEGOs, believe it or not.

Just one more — and perhaps the funniest example — eMuu, a one-eyed raindrop looking creature fit for Saturday morning cartoons (now, this is something that you have to see for yourself, at the PDF linked under Resources).

## eMo on Display

Exhibitions of eMo the emotive robot head have always resulted in audiences relating to the bot on an emotional level. See eMo — and many other robotic and scientific wonders — at the ThinkTank in Birmingham, UK. Please find the full address, phone number, hours of operation, and holiday information for the ThinkTank gallery (museum) listed in the sidebar. SV


by Jack Buffington

# Getting Keyed Up

## How to Use a Standard PC Keyboard to Control Your Robot

Have you ever wanted to send complex commands to your robot without having a computer attached to it? This column will show you how to interface with a standard keyboard that has a PS2 or AT connector at the end of its cable.

Using a PC's keyboard in conjunction with an LCD screen can allow you to easily control the values of variables in your robot or to otherwise modify its behavior. Alternately, if your robot needs dozens of buttons to control it, a PC keyboard will certainly be a cheaper and easier alternative to using standard buttons.

In an ideal world, when you pushed a key on your keyboard you would receive an ASCII character code that corresponded to that key. Unfortunately, there appears to be no obvious rhyme or reason to the codes that you receive when you press a key.

Most keys follow the rule that one byte is sent when they are pressed and, if they are held down for a while, they repeatedly send that same byte. When they are released, they send two bytes. The first is a byte that signifies that a key has been released and the second is the byte that corresponds to that key. That is fairly simple and luckily, most keys follow this rule.

For the most part, the main keyboard and the number pad on the right follow this rule but the arrow keys and keys such as page up or delete don't. Those keys send two bytes when they are pressed, send packets of two bytes if they are held down, and send three bytes when they are released. Things get even more complicated for the 'Pause Break' and 'Print Screen' buttons. The 'Print Screen' key sends four bytes when it is pressed and the 'Pause Break' button sends eight bytes.

Keyboards have sent the same data since the original PC was released, so it is likely that these oddball codes that the keyboard is sending have something to do with making it easier for those computers to process their data. Of course, these days, they make it harder for us to process their data but even still, it is relatively painless to interpret the data that keyboards send.

Let's start by looking at the physical interface between the keyboard and your processor. There are two types of keyboards that you can use: an AT keyboard and a PS2 keyboard. Both of them have the same data format. The only difference between them is the connector at the end of their cable.

There are four wires that must be connected to your project. Two of them are +5 volts and ground. The other two are clock and data. Figure 1 shows the pinouts for these two connectors. The clock and data lines are open collector I/O lines.

You might think of a PC keyboard as being an output only device, but in reality, it can also receive data from the computer. This column won't cover how to send data to a keyboard, but some things that you can tell it to do is to light up its three status LEDs, change its key repeat rate, and request that it resend the last byte that it sent.

Eleven bits are sent for each byte of data that the keyboard transmits. The signals are similar to the signals sent by SPI devices. SPI devices usually play nice and send extra clock cycles if they have a number of bits that don't fit neatly into a multiple of eight. The PC keyboard doesn't do this. It transmits 11 bits and then goes quiet. This means that you won't be able to use a standard SPI peripheral to receive its data.

Before continuing on, let's look at

***Figure 1.*** The PS/2 and five pin DIN keyboard connectors.

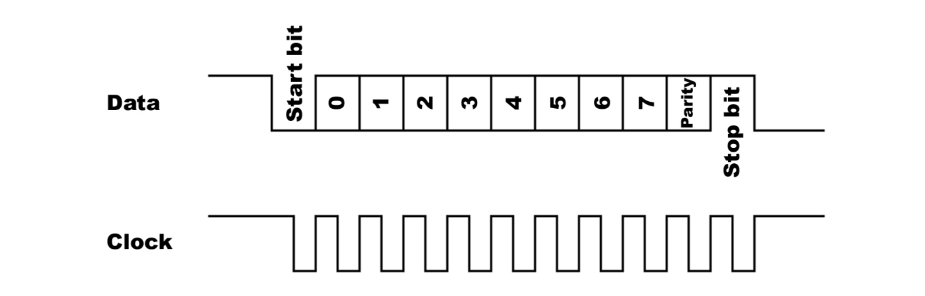



***Figure 2.*** A byte of data being sent from the keyboard.

the signals that the keyboard is sending. Figure 2 shows a single byte being sent from the keyboard to the host computer.

Some documentation says the PC keyboards are supposed to send their data with a clock that is somewhere in the range of 20 kHz to 30 kHz. The keyboard that was used for this column fell outside of that spec. Its clock rate was 13 kHz. The key point to take from this is that you can't rely on the keyboard to put out its data at a fixed baud rate. The keyboard's clock idles in a high state. Data is considered to be valid on the falling edge of the clock.

The data coming from the keyboard is as follows. The first bit is a start bit. This bit is always low. It is followed by eight bits, which are the actual data. This data is sent with the least significant bit first. The next bit is a parity bit. Specifically, it is an odd parity bit. Parity is a simple way of helping you to verify that you received your data correctly. It can allow you to detect single bit errors. If you are using odd parity, the sum of the 1s in the byte plus the parity bit will be an odd number. With even parity, the sum of the 1s in the byte plus the parity bit will be an even number.

| Byte | Even parity | Odd parity |
|---|---|---|
| 01010110 | 0 | 1 |
| 11001011 | 1 | 0 |
| 00000001 | 1 | 0 |
| 00000000 | 0 | 1 |

***Figure 3.*** Examples of parity.

Getting back to how to receive data, look at the chunk of code in Figure 4 that shows one way to receive keyboard commands. This code is pretty wasteful of your processor, though, since it hogs 100% of it. Still, it lays the groundwork for the next method that will be shown. In neither version is the parity bit used. If you are experiencing problems with the data that you are receiving, you might want to put in a check to make sure that the parity is indeed correct.

The code in Figure 4 works, but there is a better way to collect this data. Since the keyboard is supplying the clock pulses, we can connect the clock line to an I/O pin on your processor that can generate an interrupt when that pin changes. This particular piece of code uses the PIC's port B interrupt which happens whenever a pin that is set to be an input on port B pins 4 through 7 changes state. Since we only want to look at the data when the clock line has fallen, the interrupt routine simply returns when the clock is high.

The way that this interrupt routine works is that it forms a simple state machine that keeps track of which bit is being received at any given time. Each time that the interrupt happens, if the clock is falling, then it uses a switch statement to jump to code that does what is appropriate for that particular bit. It

## TECH TIDBIT

If an output is open collector, then it will only be able to pull the signal line low so you will need to have a 'pull-up' resistor to make that output actually be able to send data. A pull-up resistor is not a special type of resistor. It is simply any resistor that is connected between the positive supply (usually five volts) and the output; 10K resistors work well as pull-up resistors.

If you go higher in value, you may experience noise. If you go lower, your circuit will draw more power when the output is being driven low. While at first open collector outputs might seem like the manufacturer was just being lazy when implementing their hardware, it actually allows their device to be connected to devices that have a different operating voltage without any problems.

***Figure 4.*** Code that receives keyboard data.

```
int8 I, theByte;
int16 RXdata;

For(I = 0; I < 11; I++)
   {
   while(input(KEYCLOCK)) {}     // do nothing while the clock is high

   if(input(KEYDATA))
      bit_set(RXdata,10);
   else
      bit_clear(RXdata,10);
   rxData /=2

   while(!input(KEYCLOCK)) {}    // do nothing for the remainder of the time that
                                 //the clock is low
   } // end of for loop

// the result is now in the low byte of RXdata.
theByte = *(&RXdata);     // grabs the low byte
theByte = RXdata;         // another way to grab the low byte.  Since the upper bits
                          // don't fit into an 8-bit variable, they are simply truncated.
```

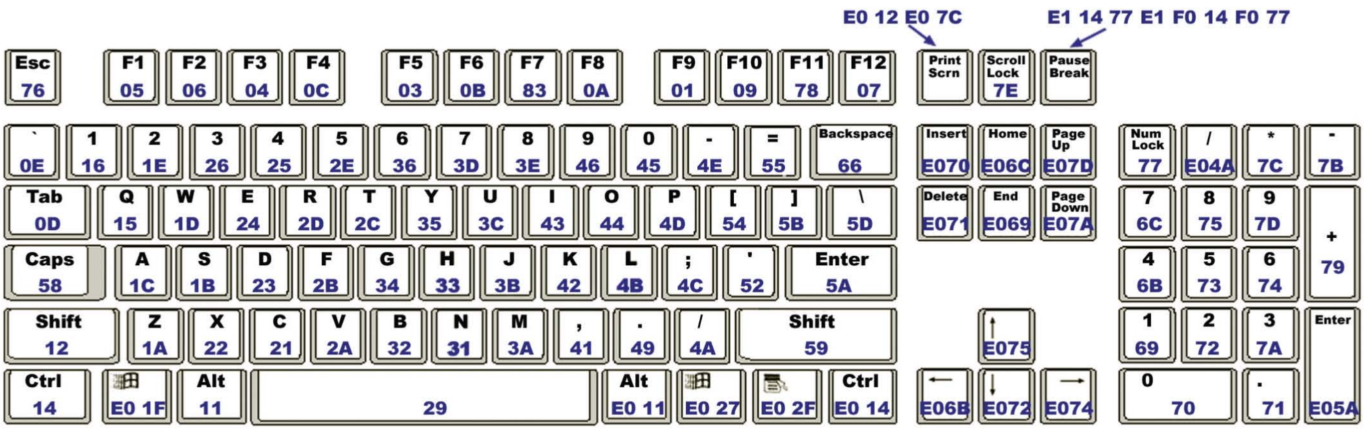



***Figure 5.*** The codes returned by each key on the keyboard.

keeps track of which bit is being operated on using a global variable. Since 8 of the 11 clock cycles load a bit into the RXbyte variable, a function was created to load the bit so that the code didn't need to be repeated eight times.

Once the byte is loaded, it is put into a ring buffer so that the main routine can process the byte when it is ready. You can download a copy of the code for this column from *SERVO*'s website (**www.servomagazine.com**) to learn more about the ring buffer. Ring buffers were covered in a previous column so that code won't be discussed here.

In a perfect world, once a byte was received you could go ahead and process it as an ASCII character, but that is not the case so you will need to do a little work on the data that you receive from the keyboard before using it. Each button on the keyboard puts out a specific character code. You might assume that it would put out different codes for capital and lower case letters, but you would be wrong. You will need to pay attention to the shift and caps lock keys yourself in order to be able to use lower and upper case letters, as well as characters such as #, $, and % that are located on keys with multiple symbols.

To convert the keys into ASCII codes, you will need to run the bytes that the keyboard sends through one of three lookup tables to figure out what the ASCII code would be. The first lookup table is for when neither the shift is pressed and the caps lock key is not pressed. This table returns the lower case letters and the numbers. The second lookup table contains ASCII codes for upper case letters and the symbols on the tops of the keys with two symbols. The final lookup table is for if the shift key is not being pressed but caps lock is on. This table has capital letters and the numbers in it.

These tables will allow you to print out characters as a typewriter would. If you want to replicate how a PC works where if caps lock is on and you push the shift key then you get lower case

## TECH TIDBIT

Interrupts allow your processor to do multiple things without too much difficulty. When an interrupt happens, the processor will stop what it was doing and jump to an interrupt routine. Some processors will jump to a specific section of code for each interrupt and other processors will jump to one general-purpose interrupt location where it will be up to you to figure out what generated the interrupt. Interrupts are usually triggered by the processor's peripherals, such as a serial port receiving data or an analog-to-digital conversion finishing. When the processor is done running the interrupt code, it will return to where it was before the interrupt happened.

letters, then you will need a fourth lookup table to deal with that. The code on *SERVO*'s website only deals with the shift key and ignores the caps lock.

As was mentioned earlier, for most keys you get one byte when a key is pressed and two bytes when a key is released. Let's use the 'A' key as our example. It sends the hexadecimal value 1C when it is pressed. When it is released, it sends two bytes F0 and 1C. The F0 signifies that the key was released. This is the way that all of the one-byte keys work.

Now, let's look at the 'Insert' key. This key sends the two bytes E0 70 when it is pressed. The E0 specifies that it is an extended key. When this key is released, it will send the three bytes E0 F0 70. It first sends the byte E0, which signifies an extended key, then it sends an F0 which indicates that a key has been released, then finally it sends the key code for Insert.

The following two keys are a lot of trouble, but if you are determined to use them, the Print Screen key puts out E0 12 E0 7C when it is pushed and will repeat that code if it is held down. When you release it, it puts out E0 F0 7C E0 F0 12. The Pause Break key is easier. It puts out E1 14 77 E1 F0 14 F0 77 only when it is pressed. It does not repeat and does not put out a code when it is released.

Okay! You now know how to read a PC's keyboard, but what can you do with it? One thing that would be really handy is if you could examine and change the values of some of the variables in your robot without having to drag your computer with you as you followed your robot. You could simply mount an LCD screen somewhere on the robot and include a keyboard connector. This is what the sample code on the *SERVO* website does.

It has an LCD screen and the keyboard connected to a PIC processor. When you type, the appropriate letter or character appears on the screen. You can connect and disconnect the keyboard as often as you want because the pull-up resistors that are on your circuit board will keep the input lines from floating.

One minor thing to keep in mind though is that each time the keyboard is attached to your robot and receives power, it will go through a power on self test and return the value of 170 (AA) to signify that it has passed. You will just need to add

***Figure 6.*** Better code to receive the keyboard's data.

```
#int_RB
RB_isr()
{
 if(!input(KEYCLOCK))
      {// this was a falling transition so go ahead and record the bit if it is a
       // data bit

       switch (whichBit)
          {
          case 0:  // this is the start bit.
             parity = 0;
             if(input(KEYDATA))
                whichBit = 0;   // this wasn't a start bit if KEYDATA was high
             break;             // this will give the code another chance to get it right

          case 1:  // these are the actual data
          case 2:
          case 3:
          case 4:
          case 5:
          case 6:
          case 7:
          case 8:
             loadBit();
             break;
          case 9:  // this is the parity bit
             break;
          case 10: // this is the stop bit.
             if(!isBufferFull())
                putInBuffer(RXbyte);
             break;
          }// end of the switch statement

      if(++whichBit == 11)
          {
          whichBit = 0;
          set_timer0(0);
          }
      }// end of if this was a falling edge
      //set_timer0(0);

}

//————————————————————————————————————————————————————————————————————————————————————————
void loadBit()
    {
    RXbyte /= 2;
    if(input(KEYDATA))
       {
       RXbyte += 128;
       parity++;
       }
    }
```

code in your robot that will ignore this value.

Having a large array of keys in a ready-made package can free you from the tedious task of wiring buttons and figuring out how to connect all of them to your microcontroller in a way that doesn't use all of its I/O pins. In its simplest form, you could use a keyboard so that each key triggered a certain event to happen, such as a light coming on or a motor energizing to drive your robot forward.

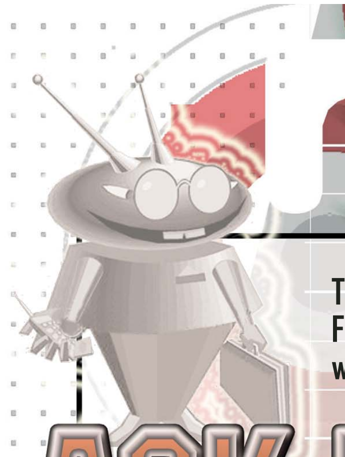
Getting more complex, if you wanted to change the values of the variables in your robot, you could just decide on a strategy where, for example, the Q key might increase a variable by one and the A key would decrease it by one.

```
// These are ASCII codes if the shift key is not pressed
const int8 noShift[] =
          {  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  9, 96,  0,
             0,  0,  0,  0,  0,113, 49,  0,  0,  0,122,115, 97,119, 50,  0,
             0, 99,120,100,101, 52, 51,  0,  0, 32,118,102,116,114, 53,  0,
             0,110, 98,104,103,121, 54,  0,  0,  0, 97,106,117, 55, 56,  0,
             0, 44,107,105,111, 48, 57,  0,  0, 46, 47,108, 59,112, 45,  0,
             0,  0, 39,  0, 91, 61,  0,  0,  0,  0, 15, 93,  0, 92,  0,  0,
             0,  0,  0,  0,  0,  0,  8,  0,  0, 49,  0, 52, 55,  0,  0,  0,
            48, 46, 50, 53, 54, 56, 27,  0,  0, 43, 51, 45, 42, 57,  0,  0};

// these are ASCII codes if the shift key is pressed
const int8 shift[] =
          {  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  9,126,  0,
             0,  0,  0,  0,  0, 81, 33,  0,  0,  0, 90, 83, 65, 87, 64,  0,
             0, 67, 88, 68, 69, 36, 35,  0,  0, 32, 86, 70, 84, 82, 37,  0,
             0, 78, 66, 72, 71, 89, 94,  0,  0,  0, 77, 74, 85, 38, 42,  0,
             0, 60, 75, 73, 79, 41, 40,  0,  0, 62, 63, 76, 58, 80, 95,  0,
             0,  0, 34,  0,123, 43,  0,  0,  0,  0, 15,125,  0,124,  0,  0,
             0,  0,  0,  0,  0,  0,  8,  0,  0, 49,  0, 52, 55,  0,  0,  0,
            48, 46, 50, 53, 54, 56, 27,  0,  0, 43, 51, 45, 42, 57,  0,  0};
```

***Figure 7.*** Lookup tables to convert the codes returned by the keyboard into ASCII codes.

Of course, if you were really ambitious, you might decide to write a command interpreter for your processor so that you could give it commands such as 'set maxspeed = 57'. That might be a bit of overkill, though.

Having a keyboard port on your robot may be a solution to input problems that you are having that won't add much cost or code space to your project. Stay tuned for next month when this column will go over how to use the new Nordic Semiconductor chip that allows for wireless data transfers of up to 1 Megabit per second! **SV**



Our resident expert on all things robotic is merely an Email away.

**roboto@servomagazine.com**

**Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?**

# ASK MR. ROBOTO

by
**Pete Miles**

**Q**. First off, I love your "Ask Mr. Roboto" section of *SERVO*. It's usually the first thing I turn to when I get my new *SERVO Magazine* each month.

I see that you've used some Maxon motors for demonstrating the wheel mounting hubs in the February column. When I saw those, I got very excited because I've been trying to find a place that sells Maxon motors for a line follower I'm building for a competition. I've tried contacting Maxon twice asking where I could buy some, but haven't gotten a response.

Anyway, do you remember where you got that Maxon motor you show in the pictures? Thanks a lot!

**— Jared Bayne**

**A**. First off, for those people that don't know about Maxon motors (**www.maxonmotorusa.com**), they are some of the highest quality DC motors and gear heads that money can buy. Their motors are extremely efficient, quiet, and powerful. They can be customized to meet *any* application that you have. Because of this, they are also some of the most sought after motors for hobbyist robots. Unfortunately, they tend to be more expensive than the budget motors that most hobbyists use in their robots. There are not very many available at surplus motor suppliers, which is one of the reasons that surplus Maxon motors are still rather expensive when compared to other surplus motors.

One of the myths that surround Maxon motors is that they don't like to sell motors in single or small quantities. This is just not true. They will sell their motors to anyone in any quantity that they want, though there are some procedures that need to be followed to order a motor. Because they have literally thousands of motor combinations, you need to know exactly which motor you want to buy before contacting Maxon.

The best place to start is to get a copy of their catalog or download their data sheets from their website. From the data sheets, select the motor, gear head, and other options that you want on the motor to determine the exact part number to order. Then call Maxon on the telephone and ask for the sales department to order the motor. If you make inquires via the Internet, you may not get any reply.

**Figure 1.** 6 mm shaft Lynxmotion planetary gear head motor and a 4 mm shaft Maxon planetary gear head motor.


If you don't know which motor to order, call Maxon and ask for the technical support, and the people there will help you select the right motor and tell you the part number for the motor and then will send you to the sales department.

Since they have so many different motor combinations, many of them are made to order, so it will take several weeks to get. But they do have some stock motors that can be obtained much quicker. These motors are identified as "Stock Program" motors. The "Standard Program" and "Special Program" take a lot longer to obtain. So, if you need the motors right away, select the motors from the "Stock Program" columns in the data sheets.

Now, with all that said, the motor that I showed in the February '06 issue of *SERVO* was obtained from Servo Systems Company (**www.servosystems.com**) a couple years ago. Servo Systems is a high-end (i.e., industrial) robotics parts supplier and systems integrator. Not only do they sell robot parts, but they sell complete industrial robotic systems, as well.

Surplus motors are not their main business, but they have one of the greatest selections of high quality surplus motors I have seen anywhere. Their website shows only a portion of

what they have to offer. Their paper catalog shows all of the motors they have. To see the surplus motors they have, you will have to dig around a bit. First, go to the "bargains and close outs" link, then go to the "Servo Motors and Amplifiers" link for DC motors or the "Stepper Motors and Drives" link for stepper motors.

The particular motor I showed in the article was a Maxon model 2322.946 precision geared DC motor with dual shaft (one shaft for an encoder). This is a 9V motor with a 120:1 planetary gearbox. At 9V, the no-load speed is 72 RPM and draws 180 mA. When I purchased this motor a couple years ago, it cost $59 and according, to their web page, it is still the same price.

If you are not dead-set on Maxon gear motors, you might try looking at the planetary gear motors from Lynxmotion (**www.lynxmotion.com**). They sell a similarly sized planetary gear motor with a 4 mm shaft for $15.25. I have some of their 6 mm diameter shaft planetary gear motors, which are extremely quiet and quite powerful. Figure 1 shows a relative size comparison of the two motors that I have.

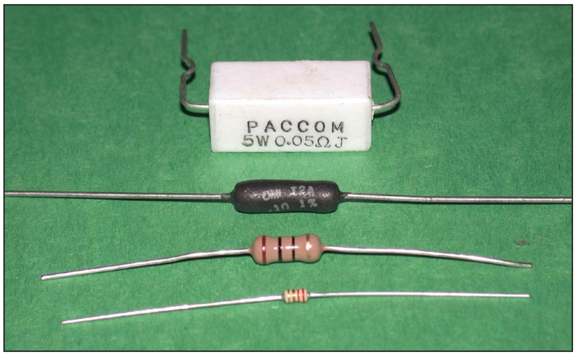



**Figure 2.** Various power rating current sensing resistors: 5W, 1W, 1/4W, and 1/8W.

**Q.** What is a good rule of thumb for choosing the right resistor to measure the current draw from a motor so that I can make sure that I don't burn out my motor controller?

**— George Stein**

**A.** There are three guidelines to consider when selecting the size of the current sensing resistor: the voltage drop that will occur across this resistor must be small enough not to have any adverse effects on the system's performance; the power rating of the resistor must be greater than the anticipated amount of power going through the resistor; and the instrument that is measuring the voltage drop must have the resolution to be able to measure changes in the voltage.

The best way to illustrate these three points is shown in Table 1. Figure 3 shows a simple schematic of how a motor and a current sensing resistor can be wired up. This example illustrates how the selection of the current sensing resistor affects the motor in a common R/C servo and how well the voltage drop can be monitored. The motor inside a standard R/C servo is the 4.8 volt Mabuchi RF-020TH motor. This example is based on a six-volt operating condition. The internal resistance of this motor is 7.63 ohms.

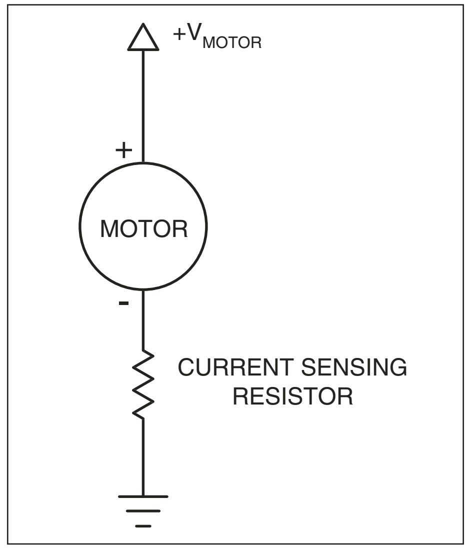
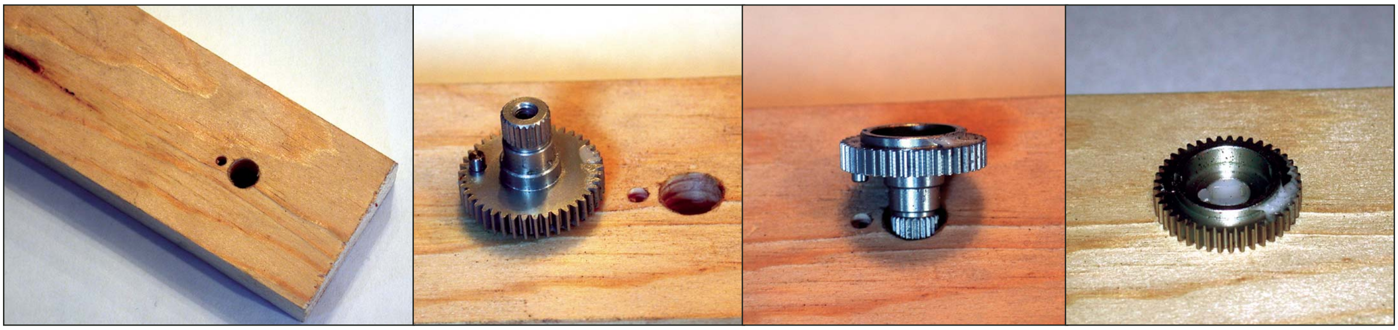



**Figure 3.** Simplified motor current sensing resistor setup.

According to the first guideline, the voltage drop across the current sensing resistor shouldn't have any adverse affects on the motor performance. From Table 1, we can see that the motor performance does change as the sense resistor's resistance increases. Under the no-load condition, there is very little change, but as the load on the motor increases (stalling being the worst case), there is a significant effect on the motor performance with higher current sensing resistance.

| | Current Sensing Resistor, ohms | | | |
|---|---|---|---|---|
| | 10 | 1 | 0.1 | 0.05 |
| Total Resistance at Stall, ohms | 17.63 | 8.63 | 7.73 | 7.68 |
| Stall Current, amps | 0.340 | 0.695 | 0.776 | 0.781 |
| Current Sensing Voltage, Volts | 3.403 | 0.695 | 0.078 | 0.039 |
| Power Dissipation, Watts | 1.158 | 0.483 | 0.060 | 0.031 |
| Minimum Power Resistor Size | 2W | 1/2W | 1/8W | 1/8W |
| Motor Performance @ No-Load | 90.3% | 99.0% | 99.9% | 100.0% |
| Motor Performance @ Stall | 43.3% | 88.4% | 98.7% | 99.3% |
| Minimum ADC Resolution for a 1% Measuring Resolution of the Stall Current | | | | |
| 8 bit | yes | no | no | no |
| 10 bit | yes | yes | no | no |
| 12 bit | yes | yes | no | no |
| 14 bit | yes | yes | yes | yes |

**Table 1.** Illustration of how the current sensing resistor affects the overall motor system.



**Figures 4-7.** Wood support with the holes and the gear being placed in the holes.

Thus, as the current sensing resistance increases, the maximum available torque (power) from the motor decreases.

In the second guideline, *the power rating of the resistor must be greater than the worst case power dissipation the resistor will go through*. Otherwise, the resistor will get very hot, its resistance will increase as it heats, and thus, cause it to get even hotter resulting in false measurements and eventual failure of the resistor. The higher the power rating of the resistor, the larger the physical geometry of the resistor becomes.

The third guideline previously mentioned is often not properly considered when selecting current sensing resistors. Often, people think about the worst case situation and feel that the resolution of their analog-to-digital converters (ADC) is sufficient to measure the voltage drop across the resistor. But this is not the typical operating condition of the motor. Under a no-load condition, the motor's current draw is typically around 5% of the stall current. Thus, the resolution of the ADC must be less than 5% of the voltage drop across the current sensing resistor at the stall condition. As a minimum, the resolution should be about 1% of the worst case stall current.

Table 1 shows four different (8, 10, 12, and 14 bit) resolution ADCs that are commonly used to monitor voltage differentials. Using five volts as a reference voltage for the ADC, this table shows which ADCs are capable of obtaining a 1% resolution of the full scale voltage range of the motor from no-load to stall conditions. Note the 14 bit ADC was the only one that was capable of measuring all four current sensing resistor test cases, but 14 bit ADCs are more expensive than the 12, 10, and 8 bit ADCs. Sometimes, the currently-available ADC resolution ends up driving the actual selection of the current sensing resistor, and motor performance is allowed to be reduced because of it.

Creating a table like the one shown helps you select the best current sensing resistor for your application. The values will be different based on which motor you are actually using. All you need to know about the motor for the table is the no-load current, stall current, and the internal resistance of the motor (though the internal resistance can be calculated based on the stall current and the voltage drop across the motors at stall). The rest of the calculations are made based on Ohm's Law.

**Q.** I am trying to turn a couple of Hitec HS-645MG servos into a pair of high torque drive motors for my robot. I tried using a screwdriver and a hammer to knock out the steel pin so it can rotate 360 degrees, but ended up wrecking some of the teeth so the gear doesn't completely turn around and occasionally the motor stalls. Is there an easy way to remove the pin out of the last gear on a Hitec 645 servo without damaging the gear teeth?

**— Jim Kemp**

**A.** The key to making sure that the gear teeth do not become damaged during the pin removal process is to make sure that

**Figure 8.** You will need to use a 1/16th-inch diameter pin punch.

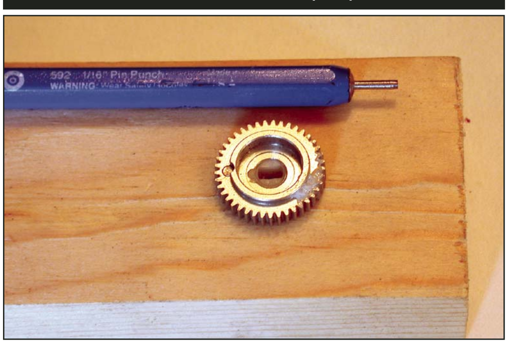


**Figure 9.** How to place the pin punch on the center of the pin.

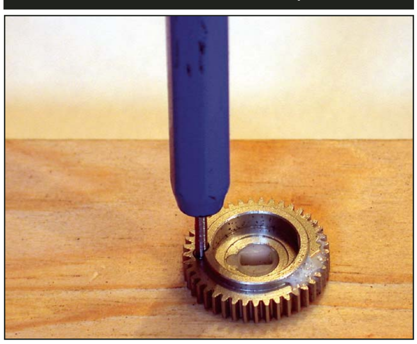
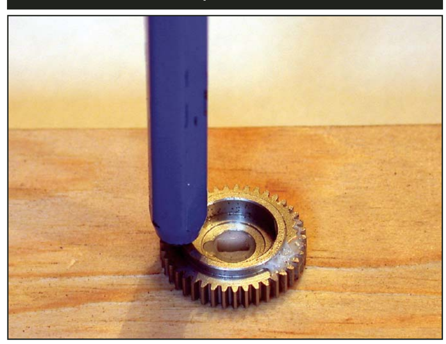


**Figure 10.** The punch should fall into the hole where the pin was knocked out



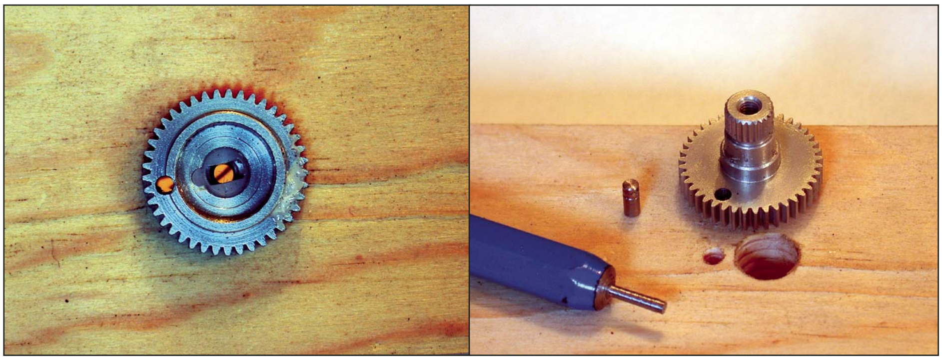
the entire face of the gear is fully supported so that the force required to remove the pin doesn't end up causing the gear to rotate sideways, which ends up damaging some of the gear teeth, or worse yet, bending the gear.

An easy way to do this is to drill a 9/32-inch diameter hole through a piece of wood. Then, using a black ink marker, color the top of the pin black. Then, put the shaft of the gear in the hole and press down on it with your fingers. This will cause the ink on the pin to mark the wood surface. Remove the gear and, using a 3/32-inch diameter drill, drill a hole through the wood at the mark. Make sure that you drill through the piece of wood so that you can retrieve the pin after you punch it through the gear. Figures 4 through 7 show the wood support with the holes and the gear being placed in the holes.

Next, you need to use a 1/16-inch diameter pin punch (see Figure 8). The diameter of the punch must be smaller than the diameter of the pin or you will end up damaging the gear. Pin punches can be obtained at just about any hardware, automotive, or tool supply store.

Place the pin punch on the center of the pin (see Figure 9). Make sure that the punch is centered on the pin to be removed, or you can damage the sides of the hole where the pin was. Then, using a small hammer, lightly tap the end of the punch a couple times. It does not require a lot of force to tap the pin out. The punch should fall all the way into the hole where the pin was knocked out (see Figure 10).

Figures 11 and 12 show the pin removed from the gear. This entire process, including making the gear support, will take about 15 minutes to make and just a few seconds to knock out the pin. This same procedure can be used to remove pins from most things that need to have pins punched out. SV



**Figures 11-12.** These photos show the pin removed from the gear.

# EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

April and May are easily the busiest months of the year for robot competitions. It's not unusual to have several different competitions happening on the same day. There are so many this year, I'll keep my comments here to a minimum so we can squeeze in as many events as possible. Enjoy!

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

**— R. Steven Rainwater**

## April

**1** **Penn State Abington Mini Grand Challenge**
*Penn State Abington, Abington, PA*
In this event, autonomous outdoor ground robots compete for a $400 prize by navigating around the campus, both on and off-road, avoiding obstacles.
**www.ecsel.psu.edu/~avanzato/robots/contests/outdoor/contest05.htm**

**1** **Penn State Abington Fire Fighting Robot Contest**
*Penn State Abington, Abington, PA*
Regional for the Trinity Fire Fighting contest.
**www.ecsel.psu.edu/~avanzato/robots/contests/outdoor/contest05.htm**

**8-9** **Battle Beach**
*Volusia County Fairgrounds, Deland, FL*
Radio-controlled vehicles destroy each other in Florida.
**www.battlebeach.com**

**8-9** **Trinity College Fire Fighting Home Robot Contest**
*Trinity College, Hartford, CT*
The well-known championship event for fire fighting robots.
**www.trincoll.edu/events/robot**

**21** **Carnegie Mellon Mobot Races**
*CMU, Pittsburgh, PA*
The traditional Mobot slalom and MoboJoust events.
**www.cs.cmu.edu/~mobot**

**21-22** **National Robotics Challenge**
*Veterans Memorial Coliseum, Marion, OH*
In addition to sumo and maze solving events, this student competition includes two unusual ones: a robotic workcell event and a pick-and-place event.
**www.nationalroboticschallenge.org**

**21-22** **RoboRodentia**
*California Polytechnic State University, San Luis Obispo, CA*
A micromouse-like maze navigation contest for autonomous robot mice. In addition to navigating the maze, robots must pick up balls and place them in a nest.
**www.ceng-web.calpoly.edu/openhouse/saturday.php**

**22** **CybAiRBot**
*Poznan, Poland*
This robot competition and exhibition will feature robot sumo played by the traditional Japanese rules.
**www.sumo.put.poznan.pl/**

**22** **UC Davis Picnic Day MicroMouse Contest**
*University of California, Davis Campus, CA*
Standard micromouse contest.
**www.ece.ucdavis.edu/umouse**

**23** **Dominican Republic National Robots Competition**
*Santo Domingo, Dominican Republic*
Robots must locate radio and light-emitting beacons. Robot builders must type in the URL. I'm not sure which task is harder.
**www.indotel.gov.do/(uyxlzr55lghapq45ntdf2245)/concursos_article.aspx?article=297**

**26-30** **BattleBotsIQ**
*Coconut Grove Expo Center, Miami, FL*
Student-built radio controlled vehicles destroy each other in Florida.
**www.battlebotsiq.com**

**26** **Istrobot**
*Slovak University of Technology, Bratislave, Slovakia, EU*
This one includes line following, IEEE Micromouse, mini sumo, and free style events.
**www.robotics.sk**

**27-29** **FIRST Robotics Competition**
*Atlanta, GA*
National Championship for the regional FIRST winners.
**www.usfirst.org/**

**27-28** **HISPABOT & Alcabot**
*University of Alcala, Madrid, Spain*
Sumo, maze solving, and Alcafutbol (soccer).
**www.depeca.uah.es/alcabot/hispabot2006/**

**28** **Robotica: National Festival of Robotics**
*University of Minho, Portugal*
The festival continues through May 1st and includes events for small autonomous robots that include RoboCup soccer and robot dancing.
**www.robotica2006.dei.uminho.pt/robotica2006**

**28** **RobotRacing**
*University of Waterloo, Ontario, Canada*
Autonomous cars must race head-to-head on outdoor courses. A two car drag race on a 20 meter straight course is followed by a multi-car, multi-lap race on a 150 meter circuit course. The circuit course is bounded by orange cones and GPS waypoints are provided.
**www.robotracing.org/**

**29** **The Tech Museum of Innovation's Annual Tech Challenge**
*San Jose Civic Auditorium, San Jose, CA*
A different robot challenge is designed each year. Check the rules on the website for the details of this year's challenge.
**http://techchallenge.thetech.org/**

## May

**6** **CIRC Central Illinois Bot Brawl**
*Peoria, IL*
Includes several classes of autonomous sumo and remote-control vehicle destruction.
**www.circ.mtco.com/competitions.htm**

**6** **LVBots Challenge**
*Advanced Technologies Academy High School, Las Vegas, NV*
Line following, line maze solving, and mini sumo, all for autonomous robots.
**www.lvbots.org**

**10** **Micro-Rato**
*University of Aveiro, Aveiro, Portugal*
Micro-Rat competition (similar to micro-mouse, but larger).
**http://microrato.ua.pt/**

**13** **Atlanta Robot Rally**
*Southern Polytechnic, Marietta, GA*
Open Contest — contestants choose their own goal for their robot. Vacuum Contest — autonomous household vacuuming contest/Mini Sumo.
**www.botlanta.org/Rally/**

**13** **RoboFest**
*Lawrence Technological University, Southfield, MI*
The RoboFest includes many events, such as LEGO robot competition, LEGO robot exhibition, advanced robot competition, and advanced robot exhibition.
**http://robofest.net/**

**13** **Western Canadian Robot Games**
*Alberta, Canada*
Robot sumo (western rules), mini sumo, walking robot triathlon, robot art contest, Atomic Hockey, and a full set of BEAM events.
**www.robotgames.net/robot_games.htm**

**19-20** **Micro Air Vehicle Competition**
*Brigham Young University, Provo, UT*
Surveillance and endurance events for MAVs. Also includes an ornithopter competition and design competition.
**www.et.byu.edu/groups/wwwmav/Tenth_MAV_Site/**

**20-21** **Mechwars**
*Eagan Civic Arena, Eagan, MN*
Radio-control vehicles will destroy each other in Minnesota.
**www.tcmechwars.com**

**26** **NATCAR**
*UC Davis Campus, Davis, CA*
Very high-speed autonomous line following.
**www.ece.ucdavis.edu/natcar/**

# NEW PRODUCTS

## CIRCUIT BOARDS

### Expansion Boards Create New Audio/Robotics and Connectivity Opportunities

Gumstix, Inc., maker of the world's smallest full function miniature computers (FFMC), now offers a new expansion board — the roboaudio-th™ expansion board — and the return of the popular thumbstix™ expansion board in a new form factor.

The roboaudio-th expansion board provides an audio-capable robotics board with through-holes. The key advantage of this new expansion board is the combination of gumstix [PXA255-based] logic, audiostix™ [UCB1400-based] logic and analog signals at 3.3 V levels, and robostix™ [Atmega128-based] logic and analog signals using 5.0 V. These signals include power, touchscreen, PWM, A/D, signalling, and an in-system programming port.

The roboaudio-th has 0.5 mm headers. At 80 mm x 36 mm with 3+4 mounting holes, the roboaudio-th expansion board connects to the 60-pin hirose connector of either the basix or the connex platforms or may be used stand-alone. The roboaudio-th costs $59.

The roboaudio-th expansion board was originally designed for Professor Richard Vaughan and his Autonomy Lab at Simon Fraser University. They plan to build a series of life-like, long-lived robot creatures: great, small and very small. Their first custom robot project is the *Chatterboxes*: a swarm of robots that communicate using sound.

"Gordon (Kruberg) combined the robostix board with the audiostix board for us, making the initial hardware layout on a big screen in front of our team during the Robotics Summit in September 2005," said Professor Vaughan. "It was a very impressive demonstration of the flexibility of gumstix products and engineering, and we got the perfect expansion board for our project."

The thumbstix-gs expansion board offers a redesign of the original thumbstix board. USB-powered with a Type A male USB socket, the thumbstix-gs expansion board offers four TTL serial ports (three TTL serial ports when used with a bluetooth enabled Basix or Connex platform), LCD signals, and two 3.3 V PWM. At 80 mm x 20 mm with three mounting holes, the thumbstix-gs board sells for $27.50.

"The addition of these expansion boards has been in direct response to requirements of the marketplace," said Gordon Kruberg, President and CEO of gumstix, Inc. "One of the reasons that customers choose our gumstix product line is because of our ability to address a wide range of function and features through an increasing array of expansion boards."

Schematics of both the roboaudio-TH and thumbstix boards are available online in the wiki, available via the support tab at the gumstix website, as well as at **http://svn.gumstix.com/gumstix-hardware/Eagle/Gumstix**

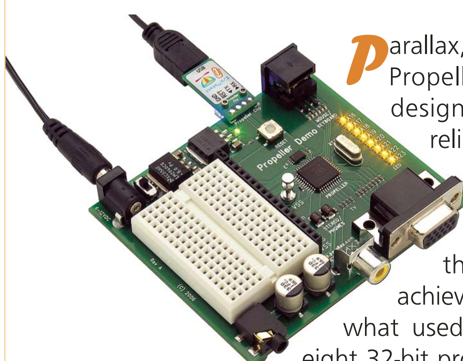
All gumstix platforms and expansion boards are available for purchase online at the gumstix website listed below.

For further information, please contact:

**gumstix, Inc.** Website: **www.gumstix.com**

## CONTROLLERS & PROCESSORS

### The Propeller Chip



Parallax, Inc. has released Propeller, a new chip designed to deliver reliability and simultaneous multi-processing. The Propeller allows the end-user to achieve with one chip what used to be done with eight 32-bit processors.

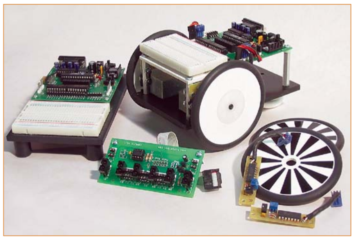
The Propeller chip was designed at the transistor level by schematic using Parallax Altera Stratix tools to prototype. Propeller is programmed in both a high-level language, called Spin™, and low-level (assembly) language. With the set of pre-built Parallax "objects" for video, mice, keyboards, RF, LCDs, stepper motors, and sensors, a Propeller application is a matter of high-level integration.

For further information, please contact:

**Parallax, Inc.** **916•624•8333** Fax: **916•624•8003** Email: **propeller@parallax.com** Website: **www.parallax.com/propeller**

## PLATFORMS

# New Low-Cost Robot Kits



The Chicago Area Robotics Group (ChiBots) has just released their club robot in kit form for sale to the general public. The CBA (ChiBots Alpha) robot kit has been in development and testing by club members for almost two years. Conceived as a low-cost standard platform to help club members get started with robotics, the kit is designed to be suitable for beginners, yet flexible and expandable to address the needs of the more experienced builder and programmer.

The CBA robot is fully controlled by its onboard BASIC Stamp® 2e or 2sx, which features protected I/O lines. Being more than just a "screw together" kit, the CBA is designed to help the builder gain the experience needed to grow in the robotics hobby.

A 60-page manual steps the builder through soldering and testing the main board, converting the RC servos to continuous rotation, and assembling the chassis. The kit is constructed using basic electrical and hand tools. A programming section of the manual along with CDs of programming software and sample programs gets the builder started.

"I own three CBA robots," said D. Kerste, past president, ChiBots. "By following the clearly written manual, I found them easy and fun to build. The CBAs are very versatile and have been used in all ChiBots games (see **www.chibot.org**). I am busy building a miniSumo version."

The CBA robot kit, complete with BASIC Stamp 2e, is priced below $100.

Wheel encoders and a uniquely flexible line-following module are also available as kits, while other add-on modules are currently being developed.

The CBA mainboard is available alone to be used in your own robot, or in combination with a breadboard, battery holder, and base to be used as a development station. A user can learn to program on the development station, then later use an upgrade kit to assemble a complete CBA. Volume discounts are offered to other clubs and organizations.

For further information, please contact:

**ChiBots Alpha (CBA)** Website: **www.budgetbot.com**

## STANDARDS COMPLIANCE & TESTING

# JAUS Testing and Training Services

CoroWare Test Labs has announced that it has developed test suites and compatibility services to provide impartial, objective conformance testing to ensure interoperability and standards compliance among intelligent, mobile service robotics applications.

CoroWare Test Labs, a leading company to test for standards compliance in the unmanned systems industry, announced availability of a suite of tools for testing JAUS/AS-4 (Joint Architecture for Unmanned Systems) compliance for both vendors and government agencies.

"As a member of the JAUS working group, as well as through our ongoing collaboration with JAUS developers, we've been active in promoting the JAUS standard and have been working for the past several months in developing a comprehensive conformance test suite," said Martin Harvey, president of CoroWare Test Labs. "We now have the capability and expertise to test JAUS compliance, giving vendors a jump-start and assuring customers that the systems they buy will interoperate with one another. This is a key step in

moving the entire unmanned systems market forward."

**CoroWare Test Lab Services**

CoroWare Test Labs offers developers and end-customers — such as government and military agencies — the following life cycle services: compliance testing, program management, test engineering, test processes, hardware and payload testing, and training.

"JAUS testing is quite complex," Harvey added. "One needs to understand the JAUS messaging fundamentals and their properties, factors such as message sequences, and message content and format; how to build this into your development life cycle; and how to test in both the lab and in the field. Our six months of preparatory labor is ready to pay off as we launch our JAUS testing services."

"From a vendor perspective, this is exactly what the unmanned systems market needs. CoroWare Test Lab's suite of tools to test JAUS interoperability allows organizations like Applied Perception to provide customers with assured compliance," said Todd Jochem, president of Applied Perception, Inc., a leading provider of JAUS-compliant unmanned solutions. "It's hard to believe, but there are already organizations who claim to provide JAUS-compliant solutions, yet have never participated in any of the JAUS community's interoperability tests. CoroWare Test Lab's test suite will allow us to clearly differentiate our products from others through independent validation."

**About JAUS/SAE AS-4**

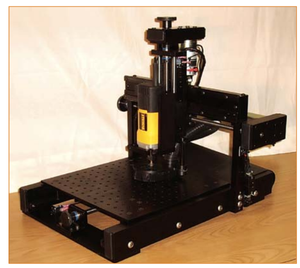
The JAUS initiative was created to develop a reference architecture that promotes open standard communications with unmanned robotic systems. The JAUS has since been transferred to the SAE AS-4 committee for commercial and military uses. The JAUS/SAE AS-4 communication specifications comprise a component-based, message-passing architecture that specifies data formats and protocols for communication among unmanned robotic systems. Moreover, JAUS/SAE AS-4 defines messages and service behaviors that are independent of any specific implementation, including computer hardware, operator use, and vehicle platforms.

For further information, please contact:

**CoroWare Test Labs**
677 120th Ave. NE #153
Bellevue, WA 98005
**800•614•CORO** Fax: **425•637•9583**
Website: **www.coroware.com**

## TOOLS & TEST EQUIPMENT

## New Desktop CNC Router



JMService is now shipping a servo powered CNC routing machine with 12 x 12 x 4 inch travels. The machine cuts decorative and precision parts in wood and plastic. It can also be used for engraving and light cutting of non-ferrous metals. The package includes an assembled machine with servo motors and an electronic controller. It includes software to program and drive the machine from any MS-Windows based computer. The machine resolution is 55,000 steps per inch, and programmed motion occurs at speeds up to 90 inches per minute. The machine includes home and limit switches and pre-configured software makes startup very fast.

The mechanism uses a fixed bridge design with a moving table. The table has convenient tapped holes at one-inch intervals to make for fast and simplified setup and work piece fixturing. Each axis has precision lead screws with antibacklash drive nuts and dual thrust bearings to eliminate uncontrolled motion. Steel bearing surfaces and guideways are used to assure precision motion and long life. The steel V-rails and aluminum components are black oxide/anodized to prevent corrosion and are made in the USA.

Optional accessories include a dust collection foot and a 3D surface scanning contact probe.

For further information, please contact:

**IMService**
P.O. Box 142
New Hudson, MI 48165
**888•451•1670** or **248•486•3600**
Fax: **248•486•3698**
Email: **imserv@imsrv.com**
Website: **www.imsrv.com**

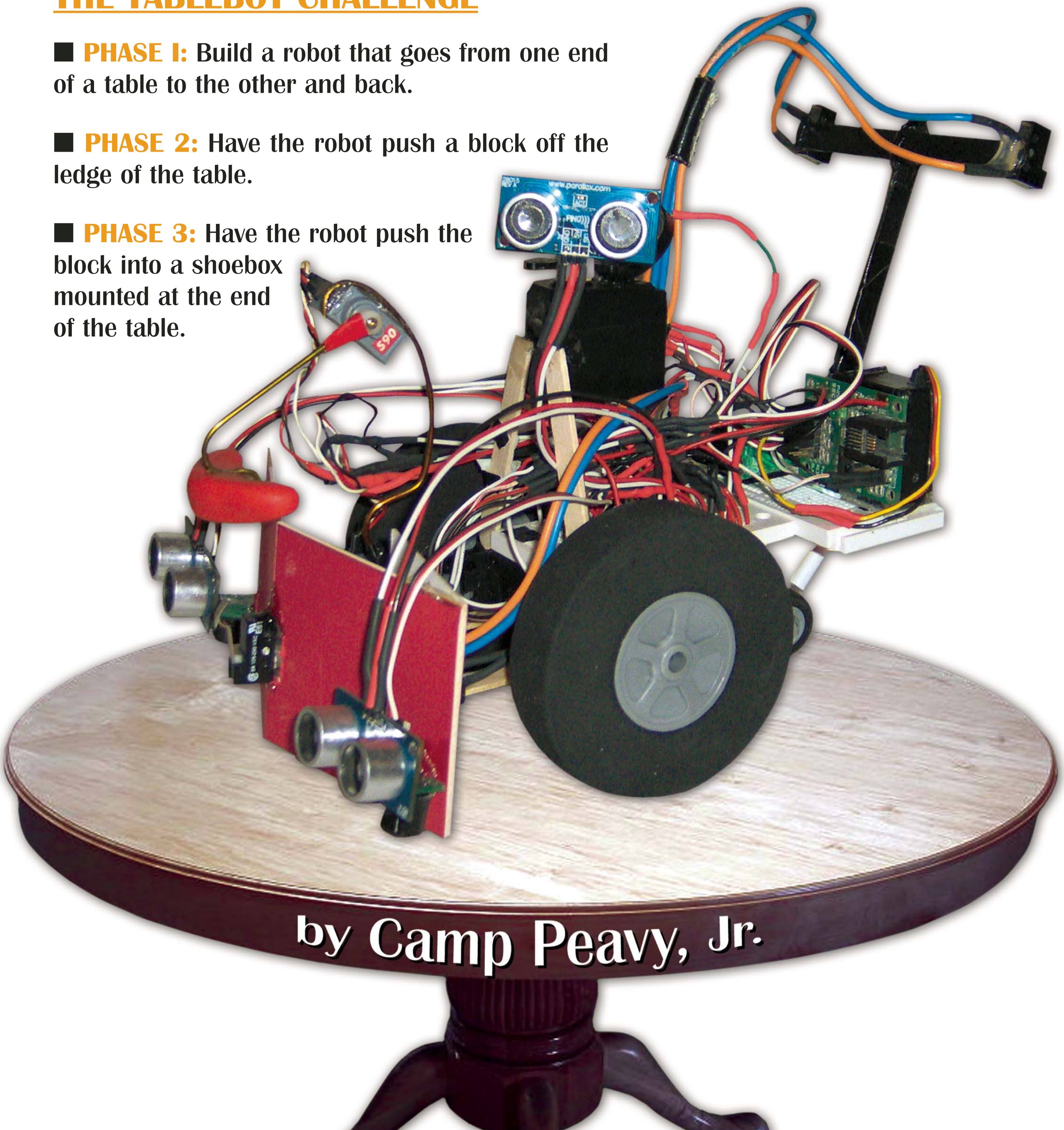
# The Ultimate TABLEBot

## THE TABLEBOT CHALLENGE

■ **PHASE I:** Build a robot that goes from one end of a table to the other and back.

■ **PHASE 2:** Have the robot push a block off the ledge of the table.

■ **PHASE 3:** Have the robot push the block into a shoebox mounted at the end of the table.



by Camp Peavy, Jr.

# The Ultimate TABLEBot

**At the April 2003 HomeBrew Robotics Club SIG (HBRC Special Interest Group), a few of us were lamenting the lack of robot building within the club. We noted building activity coincided with contests; specifically a recent line-following event and maze-busting contest called "the hurdle." We also noted that we had lots of table-top sized robots in the club including several Boe-Bots from a recent club buy. We thought in terms of what we wanted in a contest ...**

**Camp Peavy demonstrates his Phase III robot named "Buggy" at the October 2005 HBRC "TABLEBot Challenge."**

We thought we'd like something with no setup ... that is, we would like to use whatever was readily available.

At the time, we met in the Castro Middle School library and were surrounded by tables. What about table-top soccer? "Non-trivial" Bill Benson would say ... well, we could put a net around the table to catch the robots should they fall ... no ... that would involve setup ... that's when it was born ... the TABLEBot! A "TABLEBot" is defined as a robot that survives, lives, and plays on a table or pays the price.

Now, back to this non-trivial task of playing table-top soccer. Like any big job or complicated process, you break it into smaller pieces or, "phases," was the suggestion by Bill Hubbard. For the first phase, we could simply have the robot go from one end of the table to the other. Then have the robot push a block off the ledge and, finally, have the robot push the block into a shoebox mounted at the end of the table. That's it! Those are all of the rules of the TABLEBot Challenge.

There are no restrictions or limitations on the size or weight of the robot. Run what ya' brung! We don't even specify the size of the table, the block, or how you mount the shoebox at the end of the table. A TABLEBot should simply be able to survive, live, and play on a table — or pay the price.

The TABLEBot Challenge rules are purposely vague and non-restrictive so participants can use whatever robot they have (Boe-Bot, SumoBot, LEGO, etc.) and whatever table/block/box combination is readily available.

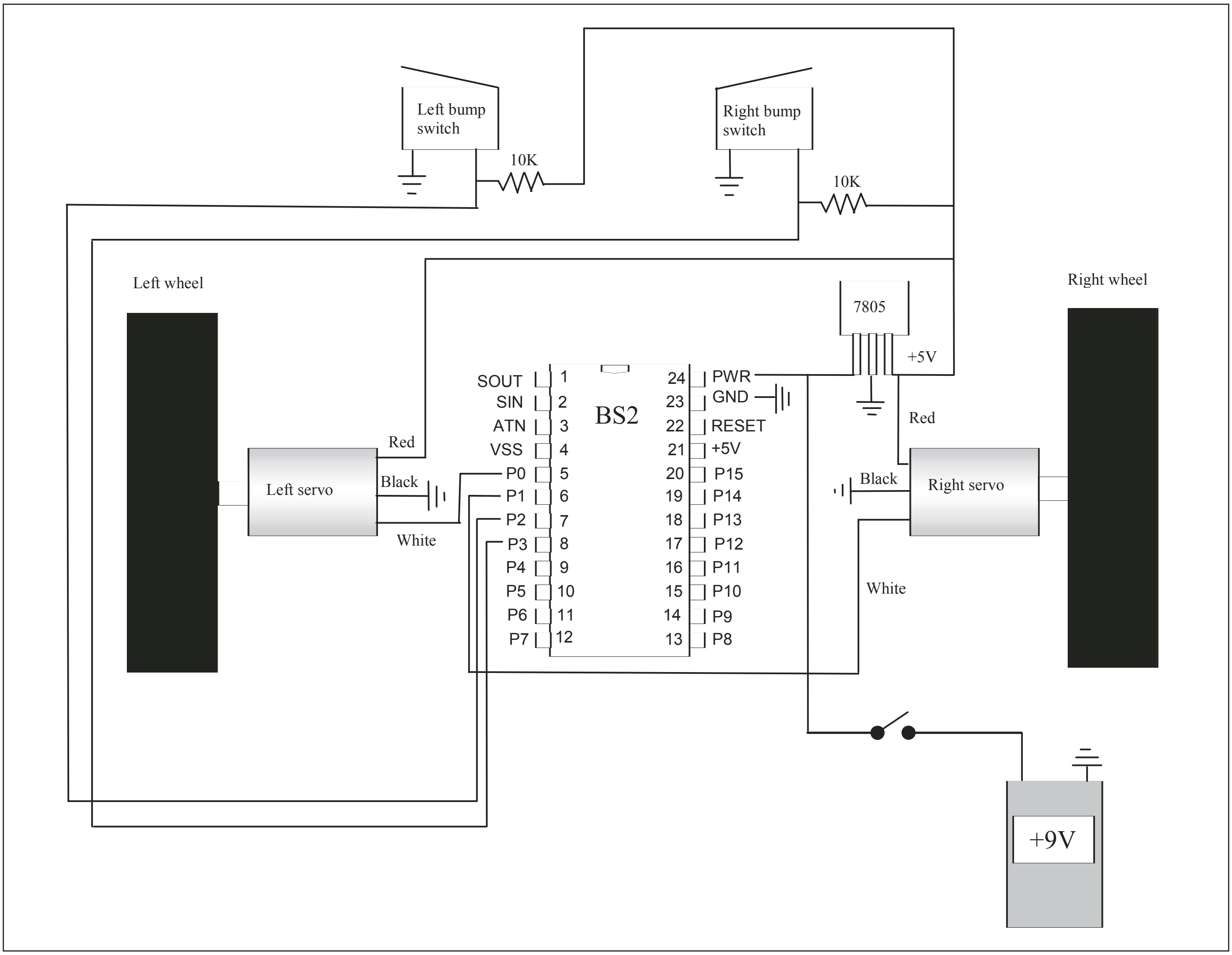
So, it was decided that we would announce the "TABLEBot Challenge" to the club at the official meeting the following week and have "Phase I" for the June meeting, "Phase II" in August, and "Phase III" in October. This would give builders two months between phases, with a relatively simple beginning and increasingly difficult stages for August and October. It was emphasized by Wayne Gramlich that this was not a "contest," but a "challenge." Ever since, it has been a joke within the club that anytime anyone refers to it as a "contest," they are immediately reprehended.

The non-competitive style of the event makes for good-natured fun without competitive pressure, including the hassle of officiating. We treat it like "show and tell" — a regular feature at the end of each HBRC meeting. Participants are always allowed to show their creations in the best light.

The "ledge" represents the real danger of the event, as relatively

**Dave Wyland anxiously watches his custom Boe-Bot as it completes "Phase I." Wyland's advice, "Never give Murphy an even break!"**

**Ted Larson of "Ologic" demonstrates "Tracker," his color following TABLEBot.**




The original PROTOBot circuit.

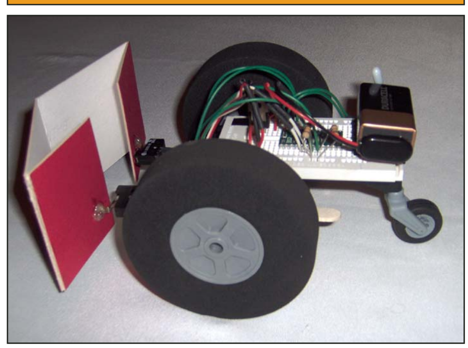
expensive robots hurl towards 30' high cliffs — 30' high to scale, that is, if the robots were full-sized cars. The ledge keeps everyone on edge! Sometimes they fall; sometimes they break. This is the reason for five-minute epoxy and this is why one of our club members — David Wyland — invented the Wyland Leash, which is a string tied around the robot to be held for debugging or if you're not completely confident of a particular environment (table color, lighting, etc). If you think about it, this is really a practical exercise for a mobile robot. To paraphrase Clint Eastwood, "A robot's gotta' know its limits ..." like say, for instance, the stairwell.

In the October '05 issue of *SERVO Magazine*, I introduced a Stamp-based educational robot called the "PROTOBot." Copies of the article are available at **www.camppeavy.com/protobot.pdf** and at the *SERVO* website (**www.servomagazine.com**). The original PROTOBot was built from common electronic components and radio-control (RC) airplane parts. It was an exercise in minimalism. I was trying to build an inexpensive, but quality robot with as few parts as possible.

In this article, we will expand the PROTOBot into the Ultimate TABLEBot. We will add downward facing ledge sensors, a block-acquisition gripper, a dual rearward tablespace detector, and more. In fact, I want to use this project to totally pimp-out the original PROTOBot and use all 16 pins on the BASIC Stamp 2 (BS2) — needed or not. The circuits in this article are from the Parallax manual or website with some modifications. The narrative describes my experiences for the edification of the community.

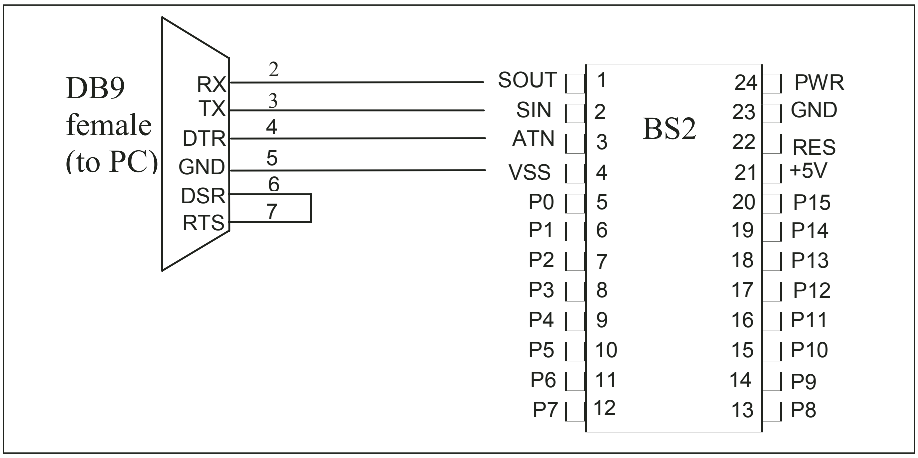
First, let's take a look at the original PROTOBot circuit. Basically, the

The PROTOBot — A Stamp-based educational robot.



```
' {$STAMP BS2}

x VAR Byte

fwd:
PULSOUT 0,1000
PAUSE 20
PULSOUT 1,500
PAUSE 20
IF IN2 = 0 THEN bwdr
IF IN3 = 0 THEN bwdl
GOTO fwd

bwdl:
FOR x = 1 TO 10
PULSOUT 0,500
PAUSE 20
PULSOUT 1,1000
PAUSE 20
NEXT
FOR x = 1 TO 3
PULSOUT 0,500
PAUSE 20
PULSOUT 1,500
PAUSE 20
NEXT
GOTO fwd

bwdr:
FOR x = 1 TO 10
PULSOUT 0,500
PAUSE 20
PULSOUT 1,1000
PAUSE 20
NEXT
FOR x = 1 TO 3
PULSOUT 0,1000
PAUSE 20
PULSOUT 1,1000
PAUSE 20
NEXT
GOTO fwd
```

**BS2 Stamp program to make PROTOBot go forward, sense obstacles, back up, and turn the opposite direction.**




**The programming cable connections from a PC to the BS2 Stamp.**

PROTOBot is just a solderless breadboard on wheels controlled by a BS2 Stamp. The program we left off with in the previous article was a module that made the robot go forward until it sensed an obstacle, at which point, it would back up, turn the opposite direction, and continue forward. We also built a homebrew cable to program the Stamp by soldering 22 gauge solid core wire to pins 2, 3, 4, and 5 and fashioning a plug to plug into the solderless breadboard.

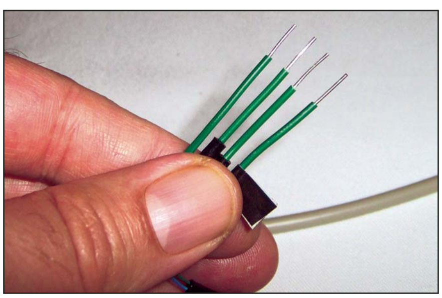
Much of the electronic construction of the PROTOBot involves simply soldering 22 gauge solid core wire to whatever you want to interface with the Stamp microcontroller. While professionally the solderless breadboard is considered a prototyping tool, it is often good enough for hobbyists.

One of my favorite adhesives is E6000 because of its strength and shock absorbency. It is commonly available in hobby and craft stores. "Goop" and "Shoe-Goo" are similar, but I really like the way E6000 sets. Another secret ingredient is five-minute epoxy — not as shock absorbent as E6000, but cures in only 10 minutes!

I made a point in the previous article that many good robots never get built because the builder doesn't have just the right screw or, perhaps, one never gets around to drilling a hole, etc., etc.. Just glue it — especially for a tabletop robot. Chances are you're going to move it anyway, so gluing gives you more options. E6000 is great for this. The mount is permanent, but if you really want to, you can peel it up, reposition, and re-glue. In prototyping, flexibility is a good thing.

The first thing to turn the PROTOBot into the Ultimate TABLEBot is something to detect the ledge of the table — after all, Phase I involves simply going from one end of the table to the other and back. I've used a variety of sensors in this challenge including mechanical and ultrasonic, but for this project, I'm going to use the discontinued SSIR (Swanson Sensor IR) sensor from Parallax. The reason I am using the SSIR is I happen to have two of them. One of the great things about having built robots for a long time is you have plenty of spare parts. It's basically an infrared LED and 38 kHz modulated receiver. The IR LED is modulated by the Stamp (FREQOUT PIN,1,38500) and the receiver pin goes low when it detects the 38 kHz signal.

On my 2005 TABLEBot "Buggy" I used two PING)))™ sensors from



**This is the homebrew plug. You may need to re-cut and re-strip the wires so that they are even. Tag or label the ground pin (5) with black tape and maintain the order of these 22 gauge solid core wires as they will be plugged into pins 1-4 on the Stamp.**


**E6000 industrial/craft adhesive — right up there with Velcro, duct tape, and five-minute epoxy.**

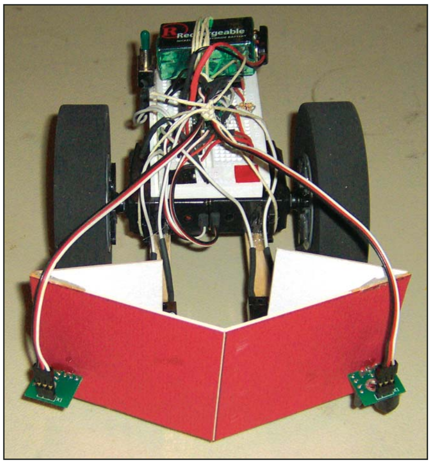
Parallax: one low, one high. Basically, the lower PING))) looked for the block and the higher sensor looked for the box. I liked the strategy and have decided to do more of the same with this project, except this time, I mounted the top or higher PING))) on a servo so it could swivel. This will help in centering the robot in relation to the box for Phase III. Plus, it looks cool!

My 2004 TABLEBot entry "Antsey" featured a servo actuated tail. Basically, I mounted two microswitches on Popsicle sticks and, before the robot reversed, it would tap down its "tail" and verify "tablespace" before going backwards. Again, it's great to have built robots for a while. Not only do you have plenty of spare parts, but actual subassemblies. This time, I borrowed Antsey's tail. But wait a minute — it's time this robot got a name!

Naming your robot has got to be one of the real joys of robotics. Sometimes you start with a name and concept and other times it just comes to you as you work. This unit will be known as "Timmay" the Ultimate TABLEBot. I like it — it has rhythm and personality!

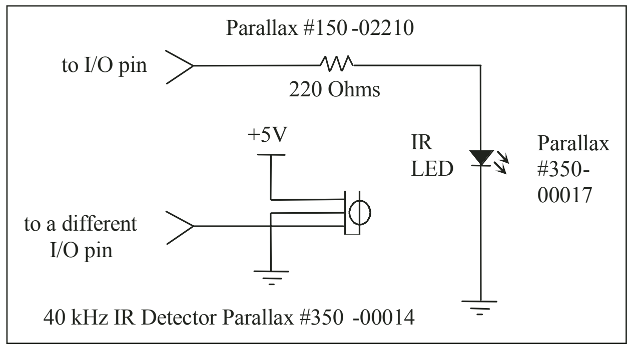
At first I tried to get away with only one ultrasonic sensor in the front, but after working through the design, I felt I needed two for differential sensing. The strategy is if the robot senses something on the left, turn right, and if it senses something on the right, turn left. If it sees something on both sides, go forward! I've added a microswitch in the center of the bumper. When it gets depressed, the gripper comes out and acquires the block. Once the block is acquired, the higher PING))) can focus on finding the box, into which the block will be deposited.

In considering the necessity of detecting the table ledge, I've elected to drive the servos from a Scott Edwards' Servo Controller board, again because I happen to have one. These boards are really cool for overcoming



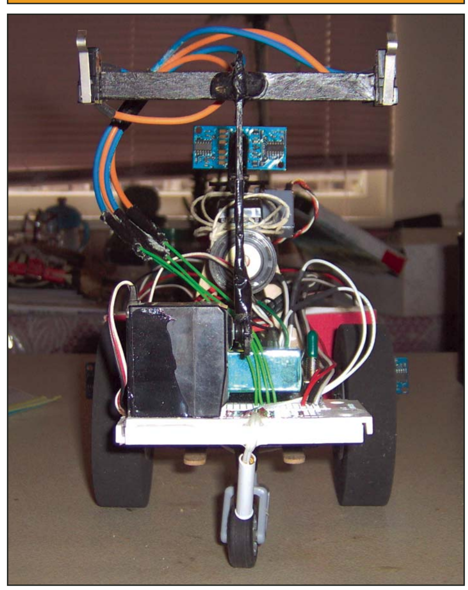
The first rule: Stay on the table! E6000 (yes, it's a verb!) the IR detectors downward to detect the ledge.


The great thing about having built robots for a long time is there's plenty of spare parts ... now, finding them is another story.

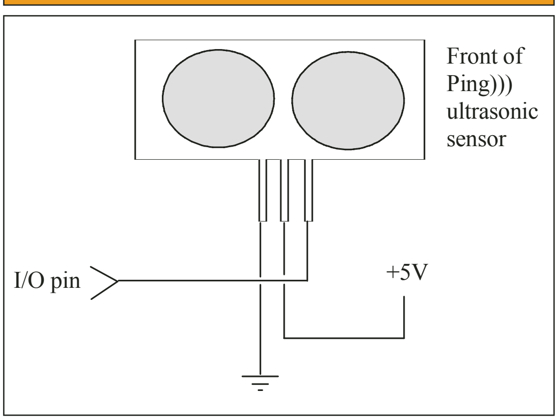



This is not the same circuit as the discontinued SSIR sensor, but does detect IR strobed at 38 kHz.

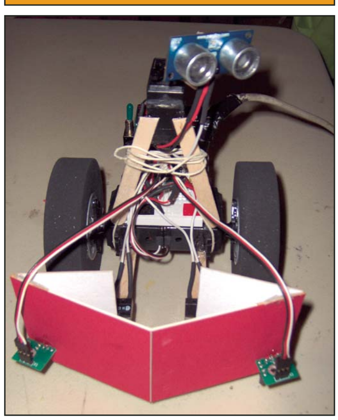
This is Timmay's rear view with the "tail" in the upward position. When the robot senses a ledge (front downward-facing IR sensors), the tail lowers and, with the two microswitches, tests for "tablespace" before reversing.

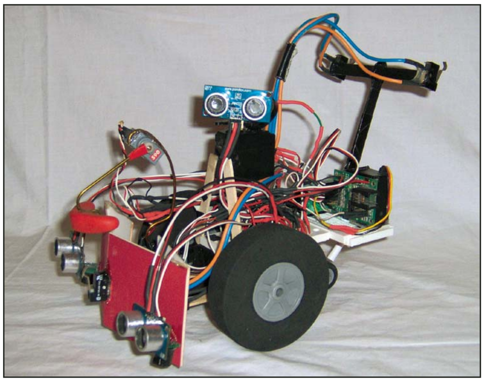


The PING)))™ is as easy as it gets: pulsout/ pulsin ... one wire operation ... the pulsin variable reads the raw time of flight.

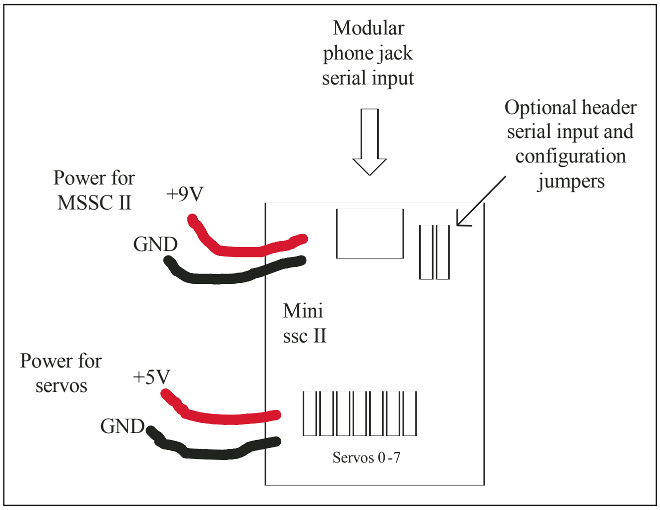



The high or upper PING)))™ ultrasonic sensor swivels on a servo to find the box.

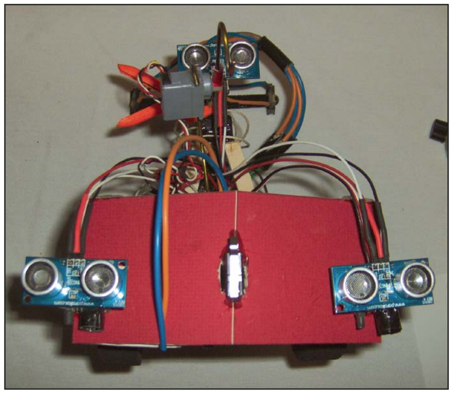




This is Timmay — the Ultimate TABLEBot. He features differential drive 3.5" wheels, PROTOBot tri-bumper, dual downward facing IR sensors for detecting the ledge, forward facing differential PING)))™ sensors for sensing the block, dual servo coat-hanger-wire arm and gripper with "Bake and Bend" Sculpy claw, palm switch for verifying block acquisition, upper level swiveling PING))) for finding the box, speaker for "beep-beep" voice, servo actuated dual rearward tablespace sensors, and reed-switch/passive caster wheel based mobility detector. Everything is controlled by a BS2 Stamp and Mini Serial Servo Controller (MSSCII). Rube Goldberg would be proud! My goal with this robot was to use all 16 pins on the BS2, whether I needed them or not. It's both a totally pimped out PROTOBot and the Ultimate TABLEBot. I still have three more Stamp ports and two more available connections from the servo controller; front and rear CdS cells and a microphone come to mind.




A specialized servo controller — like the Scott Edwards Mini SSC — helps overcome bandwidth limitations on the Stamp. Pulsing all those servos can overload the Stamp.



Here are the dual PING)))™ ultrasonic sensors. They will be used to triangulate the location of the block. The center switch will detect whether the robot has acquired the block.

bandwidth limitations on the Stamp, especially with a mission critical function like staying on the table.

This allowed me to drive six servos from one pin on the Stamp. Now, what do I do with all those freed up pins, and what of the power situation? Everything worked fine one at a time, but when I tried to flex all the muscles at once, the robot acted erratically.

Well — duh — you're using one 7805 and those servos can draw .3A each. I found a neat trick that turned out to be pure luck. Of course, the proper way to source more current is to use a bigger heatsink, but I started reading on the web about paralleling voltage regulators — I tried it and it worked! I found out later I was just lucky in that the only reason it worked was probably due to the resistance in the solderless breadboard.

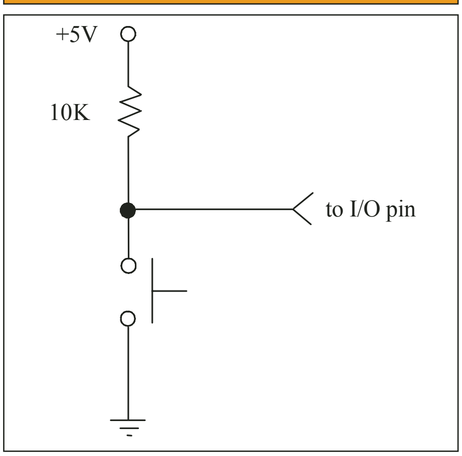
When thinking about programming robots, put yourself in the robot's place. First, the robot has to move, so pulse the wheels forward.

Then, you have to be sure that you *do not fall off the table*, so look at your downward facing IR sensors. In addition to this, you will want to "pulsout" the dual forward facing ultrasonic PING))) sensors to try and find the block. When the robot detects the block on the left, you want to turn right and when you detect the block on the right, you want to turn left. When you detect the block on both PING)))s, you want to go forward!

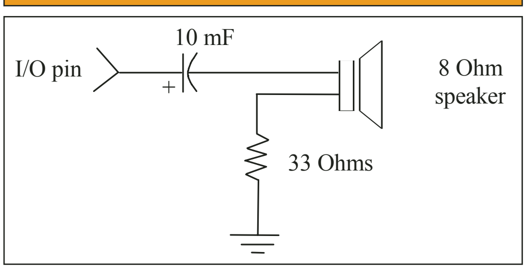
When the center switch (P11) gets pressed, the robot assumes it's the block and actuates the gripper. When the block has been acquired, the high swivel PING))) looks around for the box. When the box is located, move the block into box and you're done.

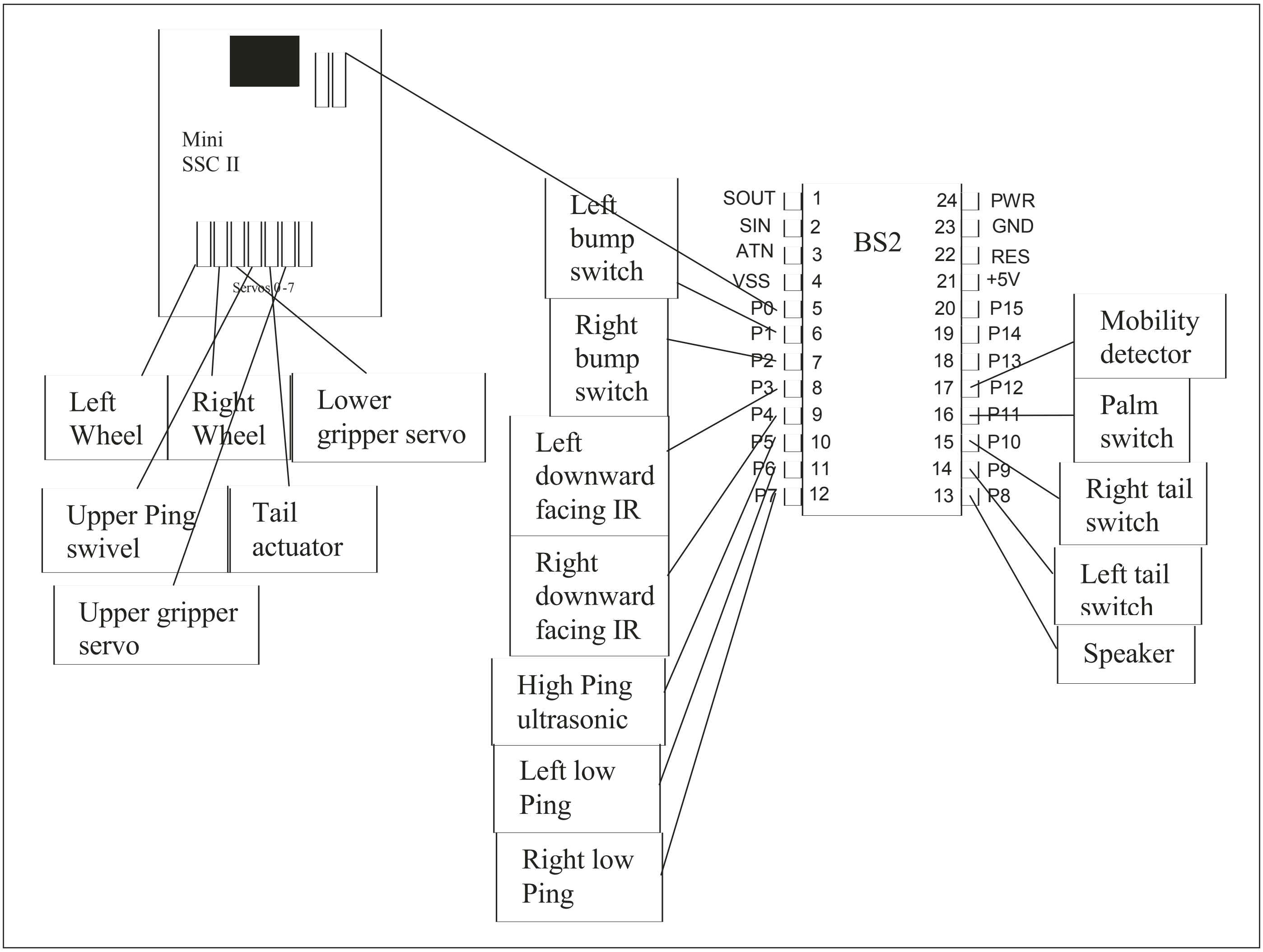
The other sensors and output devices are icing on the cake — unnecessary, but fun. All of the microswitch sensors use the standard switch interface. The speaker is useful for debugging without the debug command. Have it beep at different frequencies or inter-

This is the circuit for the bump and tail switches. It is active-low. That is, when the button is pressed, the TTL logic state goes to "0" or ground.




This is the basic speaker circuit. You could use a 40 ohm speaker in place of the 8 ohm and 33 ohm resistor. Give your robot a "beep-beep" voice.






**I'm using 12 of the 16 general-purpose I/O pins on the BS2 Stamp and six of the seven servo ports on the Scott Edwards Mini Serial Servo Controller. The current program accomplishes Phase III of the TABLEBot Challenge.**

vals for different subroutines. This way, you know what your robot's thinking.

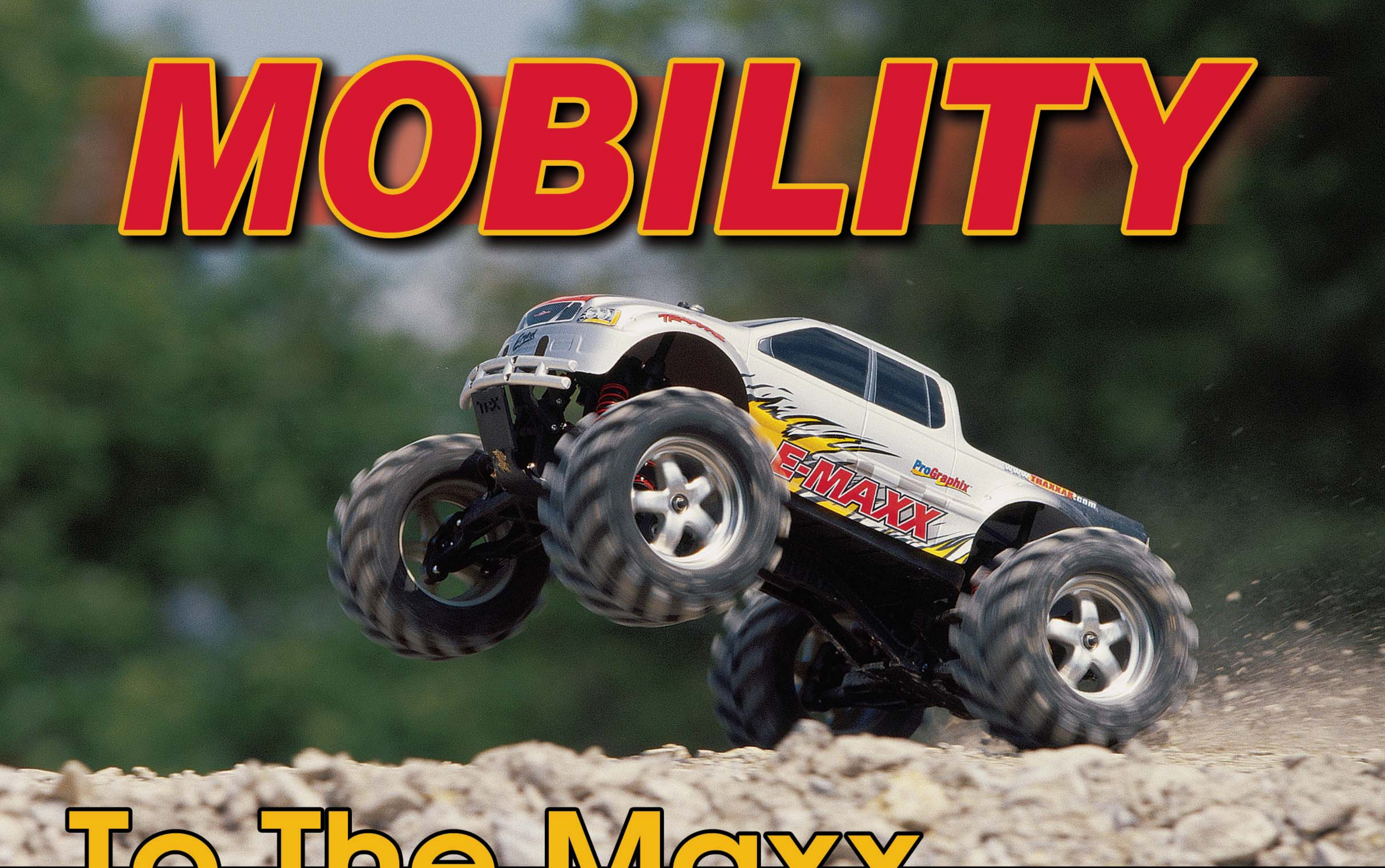
Here's a couple tips for working with the BASIC Stamp. When you encounter weird problems, check your power (buy a battery tester). The Stamp will not run when connected to the computer unless you are running debug. I assume it has something to do with the DTR connection from the computer.

Build and test, build and test, build and test. Make frequent incremental saves of your code; "Save As" and increment the number by one. Use debug commands as you debug the system and remark (') them out as you finish testing and especially before you place the robot on a "live" table. This way, if you have to test again, you can simply un-rem them.

As you drop a small part, look quickly where it's going. Otherwise, it might be lost forever. Find tools that you enjoy working with. It makes the activity more enjoyable.

In building robots — as in life — all things conspire to keep you from doing what needs to be done. Regardless, remember that building is a series of small steps. Don't think about the complexity of the overall project or you will become discouraged. Do the small things on a daily basis and eventually the whole thing will be done. If you come to a dead-end or stopping point with one subsystem (glue drying or need a part), consider what can be done on another subsystem. I think too many times we try and come up with reasons as to why things can't be done or we didn't get around to them rather then "just doing them." Enter contests! There's nothing like a deadline to force one to create.

At this point, Timmay is capable of depositing the block-in-the-box-on-a-table, but I still haven't used all 16 pins on the BS2 Stamp. Plus, I haven't integrated the "mobility detector" — a small, rare earth magnet (RadioShack #64-1895) embedded in the passive caster tire and a small reed switch (RS#49-496) glued to the solderless breadboard body. I pipe the on/off signal from the reed switch into P12 and, as long as the wheel is turning, the robot is "happy." The good news is I still have until June for the 2006 TABLEBot Challenge, Phase I. **SV**




# MOBILITY

# To The Maxx

## PART 1 — Hacking the E-Maxx

by Chris Cooper

**If you are thinking about winning the next SRS (Seattle Robotics Society) Robo-Magellan, or just looking for the ultimate four wheel drive (4WD) robotics base, then look no further than the Traxxas E-Maxx.**

The E-Maxx is the premier 1/10th scale electric 4WD Remote Control (RC) monster truck and the perfect starting point for your next mobile robot. In this article, I will point out the benefits of using the E-Maxx and highlight essential modifications to steer you down the road to building a successful mobile robot.

As Michael Helm pointed out in his June 2005 *SERVO* article "Converting Low-Cost RC Cars into Simple Autonomous Robots," modifying toy RC cars is an inexpensive and effective approach to creating simple robots. However, toy RC cars have many limitations — they cannot support additional weight, they have limited room for sensors, they do not have aftermarket parts, and they are difficult to modify.

The E-Maxx is different. A quick look at its feature set and you'll see it is no toy:

- Dual motors — The E-Maxx can reach speeds of over 30 miles an hour.
- Two speed transmission.
- Shaft drive 4WD with front and rear differential.
- Fully adjustable suspension with four inch ground clearance and 3.5" of travel.
- Ackerman steering.
- Adjustable toe-in and camber.
- 14.4 volt power system.
- Electronic speed control (forward/reverse).

With a price tag of $300 (estimated), the E-Maxx is 10 times the price of a toy RC car, but you get a high performance chassis even the most experienced robot builder would be hard pressed to match in capability and price. By using the E-Maxx, you can leverage its proven design, reduce risk, save time, and focus more attention on the task at hand — transforming the E-Maxx into an

**Photo Above:** The E-Maxx RC monster truck makes an excellent robotics base. *Photo courtesy of Traxxas.*

off-road autonomous explorer for your backyard and beyond.

## Modifying the E-Maxx

The E-Maxx is designed to be easily upgradeable and customized. In fact, the success of the E-Maxx has spawned an entire industry devoted to after-market parts and upgrades. Every part of the E-Maxx can be replaced if broken or upgraded to a higher performance part.

Before you make any modifications, familiarize yourself with the "Anatomy of the E-Maxx" on page 7 of the E-Maxx Owner's Manual (available online — see Resources sidebar) and take the truck out for a test drive to make sure it is working properly.

After a successful test drive, you are ready to make the following modifications:

⇛ Upgrade the steering servo to improve maneuverability.

⇛ Strengthen the suspension to support the additional weight of the robotic hardware and deck.

⇛ Change the transmission's gear ratio to increase the effective torque.

⇛ Choose and install appropriate batteries.

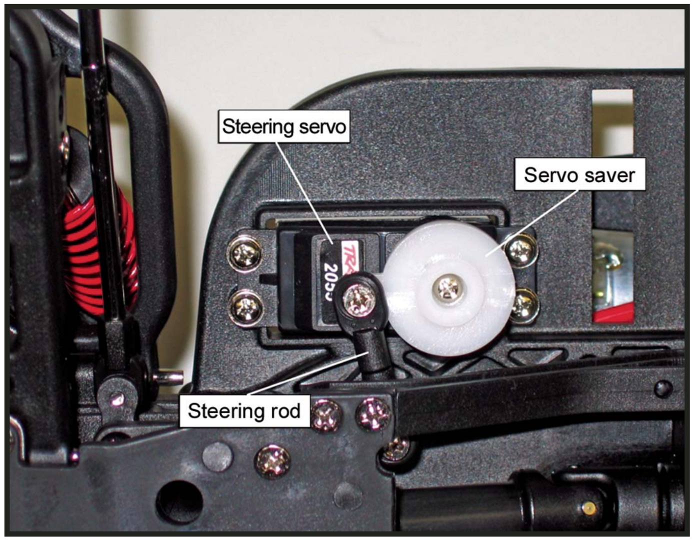
⇛ Create and mount a deck to hold the robotic hardware.

### Upgrading the Steering Servo

Mostly because of the oversized high-traction tires, the stock steering servo in the E-Maxx is inadequate for robotic applications. If a servo is too slow, steering will be unresponsive. If a servo does not have enough torque, the steering will be sloppy and imprecise and will not be able to turn the wheels unless the car is moving. Upgrading the steering servo will improve the overall control of the vehicle.

The stock servo provides about 80 oz/in of torque at a speed of 0.22 sec/60 degrees. Choosing a replacement steering servo is a balance between price and performance. For this application, torque takes a higher priority than speed. A "slow" servo might not feel as responsive as a "fast" one, but it won't limit your control of the vehicle.

**Figure 1:** The steering servo on the stock E-Maxx.




I installed the Hitec HS645MG, which, at $40, is priced less than the other servos, but supplies 133.31oz/in of torque. At 6V, this servo moves at a speed 0.20 sec/60 degrees and gives mid-range performance, though it requires a Kimbrough servo saver, which costs about $7.

The Hitec turns the front tires well under most conditions except on short "grippy" carpet, which seems to keep the wheels from turning precisely and can prevent them from traveling the full range of motion. On grass and dirt, the wheels turn quickly across their full range.

Other servo options include:

⇛ *Futaba S9402 High Speed Coreless BB* — Provides 111.10 oz/in of torque and quick movement at a speed of 0.09 sec/60 degrees at 6V. The Futaba servo costs around $72. You can reuse the servo saver off of the stock servo without having to buy a Kimbrough servo saver.

⇛ *Airtronics 94358Z ERG-VR* — Provides a whopping 200 oz/in of torque at 6V and moves at a speed of 0.10 sec/60 degrees at 6V. Be prepared to pay around $95 for this high-end performer and don't forget — you will also need to buy the Kimbrough servo saver.

Note: *The Kimbrough Servo Saver comes with adapters to fit Airtronics, Futaba, and Hitec servos. Simply drill out one of the large holes on the servo saver to allow the servo saver bolt to fit.*

## OPTIONAL UPGRADES

### Upgrading Nylon Parts to Aluminum

Every nylon part can be upgraded to an aluminum part. Aluminum increases the strength of the chassis but at a higher cost and weight.

### Tires

Stock tires can be replaced with a larger 7" diameter tire for increased ground clearance and a better ability to climb over larger obstacles. Additionally, bead-lock tires and rims can be used to decrease wheel weight and eliminate the need to glue tires to the rims.

### Motors

Brushless motors will increase motor performance and decrease the amount of power consumed. Brushless motors require special control and need to be paired with a brushless controller.

### Batteries

Additional batteries can be added to increase range and driving time at the cost of additional weight.

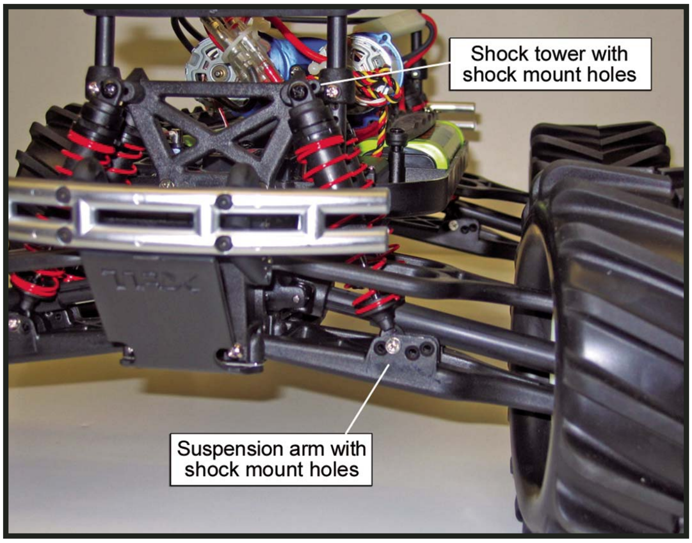



**Figure 2:** Replacing the springs on the E-Maxx.

## RESOURCES

Traxxas EMaxx Owner's Manual
**www.traxxas.com/products/electric/emaxx/trx_emaxx_views.htm**

Tower Hobbies
**www.towerhobbies.com**

MachineBus
**www.machinebus.com**

Al's Hobby Shop
**www.alshobbyshop.com**

To upgrade the steering servo:

**1.** Flip the truck upside down, as in Figure 1, and unscrew the servo saver bolt from the servo saver/steering rod. Remove the four screws attaching the servo to the chassis.

**2.** Unplug the servo connector from the receiver and note the channel number and the wire positions.

**3.** Feed the replacement servo wire through the mounting hole and plug the connector into the same position on the receiver.

**4.** Mount the replacement servo onto the chassis using the original four screws.

**5.** Before connecting the steering rod/servo saver, turn on your radio and receiver and turn the wheels left and right, then back to neutral to center the servo.

Note: *Depending on the brand of servo, the signal to rotate clockwise/counter clockwise may be reversed. Make sure that the servo saver is positioned so the wheels are straight when the servo is at center.*

If you plan on running in high brush or in rough terrain, you may also want to install a servo cover to protect the steering servo from damage. An E-Maxx aluminum servo guard costs around $20.

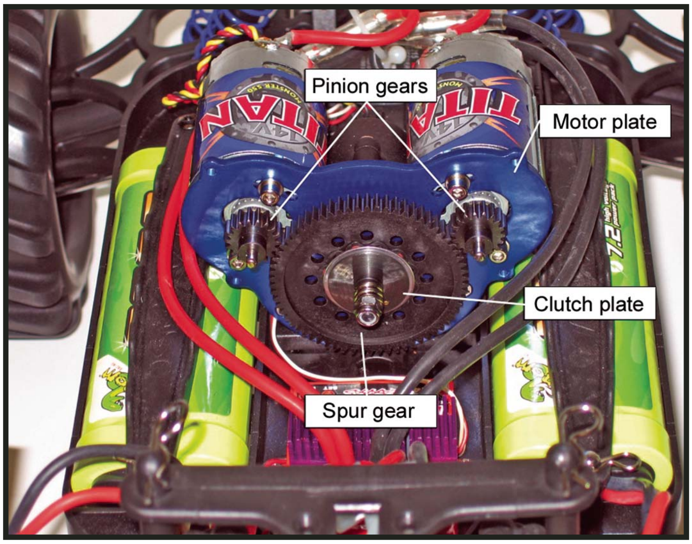



**Figure 3:** Changing the gear ratio on the E-Maxx.

### Strengthening the Suspension

The stock suspension is fine for the E-Maxx chassis. But, in order to support the additional weight of sensors and controllers, the suspension should be strengthened by replacing the shocks, using heavier weight shock oil, or replacing the springs. For more information, see "Adjusting Your E-Maxx" on page 22 of the E-Maxx Owner's Manual. The resulting suspension should be able to hold an additional 10–20 lbs.

The easiest and cleanest choice is to replace the springs with a heavier version. Two sets of four Trinity E-T Maxx XX Heavy (dark blue) springs provide enough stiffness to insure the E-Maxx will never bottom out, and they also help reduce body roll.

To replace the springs:

**1.** Remove each shock (see Figure 2) by unscrewing the top from the shock tower and the bottom from the suspension arm. The lower shock boot has a notch in it so it can be removed after slightly compressing the spring.

**2.** Remove the boot and then the spring.

**3.** Insert the new spring, replace the boot, and reattach the shock.

When you replace the upgraded shocks on the chassis, notice there are multiple mounting positions to choose from on the shock tower (0, A, B) and on the suspension arm (0, 1, 2, 3, 4). If you will be running primarily in rougher off-road terrain, you'll want a softer suspension such as 0-2. If you will be running on road, lower the ride height to B-4 to increase stability and create a more progressive suspension. The on-road settings will provide increased stiffness, which will help prevent rolling during turns, diving when braking, and squatting when accelerating. If you will be running in a variety of environments, then return the shocks to their original A-2 positions.

Make sure you have a shorty or

ratcheting offset screwdriver. If you decide to change the shock mount positions, be prepared to work a little to screw into the unthreaded holes in the shock towers. I chose to optimize the shock position for off-road use with a softer 0-2 suspension. After a lot of screwing and unscrewing to find the right mounting position, the suspension is noticeably stiffer and ready for the additional weight.

### Changing the Gear Ratio

In order to effortlessly handle the additional weight, you will want to change the gear ratio in the transmission. The E-Maxx transmission comes with a 66 tooth spur gear and two 18-tooth pinion gears. You have a number of gearing options, such as:

- Spur gears in 64, 66, 70, and 72 tooth versions.
- Pinion gears in 12-22 tooth versions.

For robotic applications, I typically decrease the top speed and increase the torque. To do this, I replaced the gears with a 72-tooth spur gear and two 12-tooth pinion gears. Given this gearing, I get an overall reduction — the number of turns the motor makes for each revolution of the tire — of 47.30 in first gear and 29.37 in second gear. See the E-Maxx Owner's Manual for an excellent gearing chart to suit your needs.

To change the gear ratio:

**1.** Unscrew and remove the plastic transmission cover to reveal the spur and pinion gears (see Figure 3).

**2.** Loosen the setscrews and remove the pinion gears.

**3.** On the spur gear, remove the lock nut, clutch spring, and outer clutch plate.

**4.** When you remove the spur gear, make sure you transfer the brown circular slipper-clutch pegs from the old spur gear to the new one.

**5.** Loosen the screws that hold the motors to the motor plate. This will allow you to adjust gear positions so the gear teeth mesh properly.

**6.** Slide the new gears on the axle and tighten the set screws.

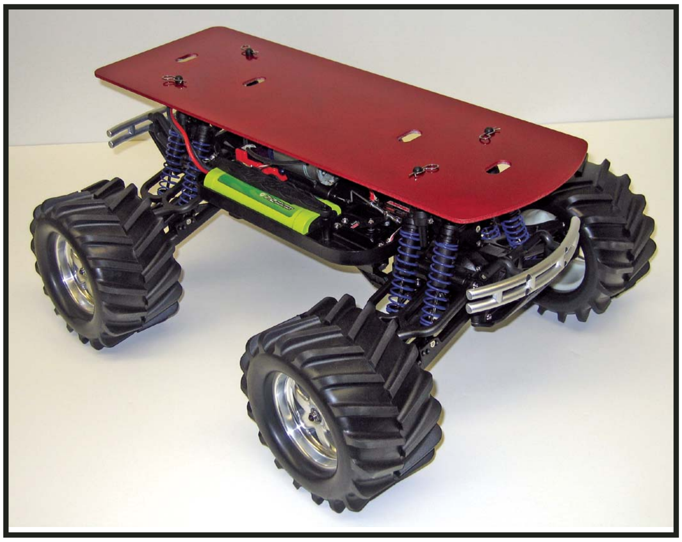


**Figure 4:** The modified E-Maxx, with an upgraded steering servo, stiffer suspension, greater engine torque, NiCd batteries, and a spacious robotics deck.

### Choosing and Installing Batteries

The E-Maxx needs two 7.2V battery packs (purchased separately) to run. They provide the motors with 14.4V of power, which is regulated by the electronic speed controller to supply 5V of power to the receiver and the two servos. Battery packs are typically made of six or seven individual Nickel Cadmium (NiCd) cells, wired together and wrapped in a plastic film. NiCad packs typically weigh around 15 oz. Battery packs are rated in milliamp hours. The higher the rating, the longer the batteries will run. Typical NiCd battery ratings are in the 2,000 mAH to 4,000 mAH range.

Lithium Polymer (LiPo) batteries are an excellent alternative to using NiCd batteries. They are extremely light, weighing about 9.1 oz. A LiPo pack can currently provide up to 4,800 mAH, but be prepared to pay over $100 a pack. You should proba-


## AUTHOR BIO

*Chris Cooper is currently a software architect for Chicago-based Machine Bus Corporation. He has a B.S. in Computer Science from the University of Illinois, has presented at the OMG's Robotics SIG on Distributed Control Systems, and is a member of the Chicago Area Robotics Group (Chibots). He can be reached at cooper@coopertechnical.com*


## PARTS LIST

*All E-Maxx parts can be found at any local or online hobby shop. I purchased mine online from Tower Hobbies and locally from Al's Hobby Shop, which also provided me with expert RC advice.*

**Steering**
Hitec HS645MG High torque servo ($40)
Kimbrough servo saver ($7)

**Suspension**
Two packs of four Trinity E-Maxx XX Heavy springs ($9)

**Gearing**
Traxxas 72-tooth spur gear ($3)
Two Traxxas 12-tooth pinion gears ($4)

**Batteries**
Two 3,300 mAH Venom NiCd packs ($24 each)

**Deck**
MachineBus MaxxDeck Aluminum deck ($40)

bly keep extra packs on hand, using two to run while another two are recharging.

Remember, you will need to get a capable charger. I've found it convenient to be able to charge more than one battery pack at a time. It's also nice to be able to charge a variety of battery types.

For less than the cost of two single chargers, the MRC SuperBrain 977 charger can handle two packs at a time. It is an AC/DC charger with dual output and discharge functions, which increases battery life. The large LCD display shows battery status, battery voltage, charge rate, peak threshold in mV, capacity in MAh, the number of cells in the pack, and charging time in minutes.

### Creating and Mounting the Robotics Deck

The E-Maxx has four adjustable posts used to mount a variety of body styles such as a pickup truck, van, or Hummer. While the truck bodies look nice, they don't provide for a great robotics base. The body is too flexible to properly support additional weight, and it is hard to find adequate space to mount sensors and electronics. Instead, by adjusting the front and back posts to be level with each other, I used a custom deck to provide enough stable space to mount the control system and all the sensors.

I considered using Lexan, hobby plywood, or aluminum in both 3 mm and 1/8th inch thickness for the deck. Lexan is affordable, light, and looks nice, but I find it can crack easily and it also holds a static charge. Hobby plywood is cheap, light, and easy to work with. You can easily cut holes and screw into it, which is convenient for testing out a variety of sensor configurations.

If you want to build an E-Maxx deck, I have posted a template at **www.machineBus.com/emaxx** which lays out the proper hole spacing. If you would rather buy than build, you can purchase a red anodized aluminum deck there, too, for $40. The aluminum deck looks great and provides for a stronger, more rigid base. It is still very light, will save you from doing any machining, and does an excellent job of heat dissipation — think giant heatsink!

## The Modified E-Maxx

You can fully modify an E-Maxx monster truck for robotic endeavors, as in Figure 4, using off-the-shelf parts and common household tools.

In my next article, I will describe how to control the DC motors, steering, and shifting servos. To put the modified E-Maxx through its paces, I will show you how to add in tele-operation through a Bluetooth wireless connection, and I'll provide code for you to drive it remotely with a PC and a joystick. **SV**

# ROBOLYMPICS

**www.robolympics.net**

## San Francisco, California: Fri-Sun, June 16-18, 2006

Robots from around the world invade San Francisco for the Third Annual International ROBOlympics!

Join us in 2006 for the only international combined-event robot competition. Compete in any of 60 different competitions, including no-cost junior league events (for kids 17 and under.) At the 2005 event, over 650 engineers from 13 different countries competed with more than 450 robots! No matter which event you compete in, you'll still meet hundreds of robot builders from around the world, see new robots you never knew existed, and learn more about robots than you imagined possible!

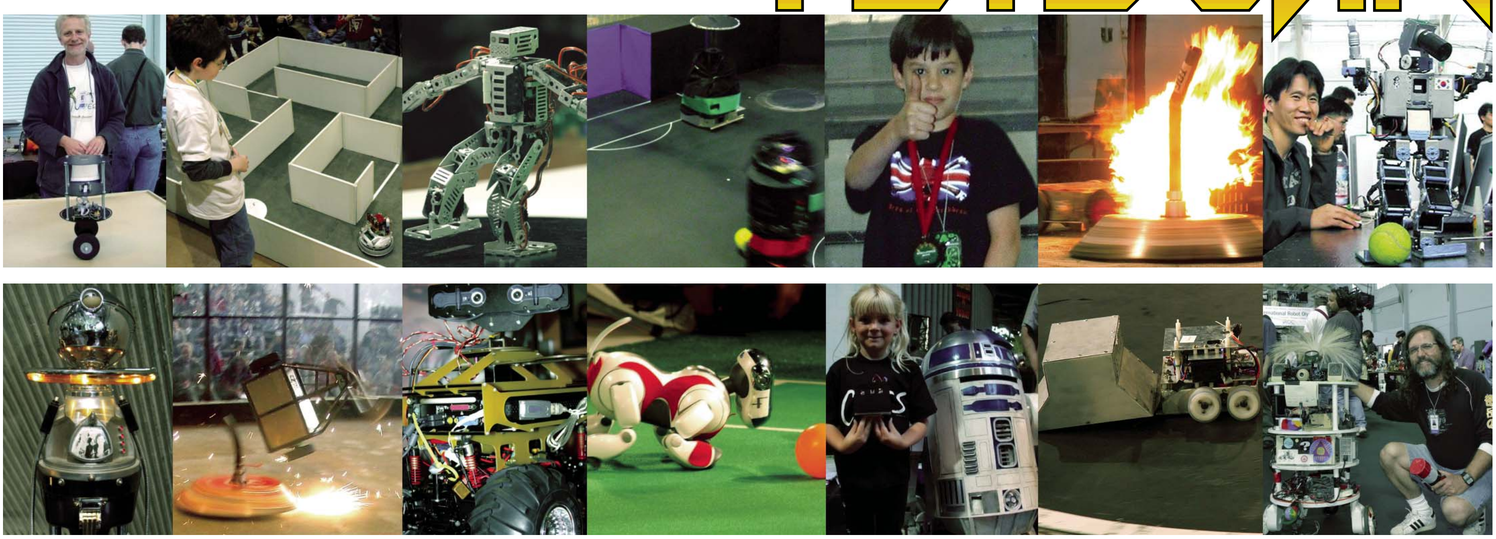
**Don't miss this show!**

Sponsorship opportunities are still available for the 3nd Int'l ROBOlympics! Take advantage of the tremendous publicity potential! Robotics is becoming the new lifestyle sport of the thinking age. The Renaissance had music, we have DC motors. ROBOlympics is the channel to a growing audience with both a voice and a high disposable income. Contact us for info on this exciting opportunity: sponsors@robogames.net

### Events at ROBOlympics:

**Combat:** 150g-340lbs and 2 autonomus classes;
**Robot Soccer:** Humanoid, MiroSot, NanoSot, RoboSot, Aibo, Small League, Middle League;
**Sumo:** 25g - 3kg;
**Open:** Robot Triathlon, Ribbon Climber, Line Slalom, Mindstorms Challenge, Best of Show, Maze Solving, Aibo Performer, Balancer Race, Fire Fighting, Mindstorms Open, Table Top Navigation, Vex Open, Vex Challenge, Biped Race, Walker Challenge, Robomagellan
**BEAM:** Speeder, Photovore, Robosapien Hacker
**Junior League:** Lego - Woots and Snarks, 120 lb Combat (Jr), Mindstorms Challenge, Best of Show, 500 g Sumo, Handy Board Ball, Mindstorms Open, BasketBall Challenge, Vex Open, Vex Challenge
**Robo-One:** Wrestling, Agility, Stair Climbing, Door Opening, The Eagle, Toss
**Art Bots:** Musical, Static, Kinetic, Painting, Bartending
**Bot Hockey:** 3kg & 12 lb
and of course:

**TETSUJIN**



**The Third International ROBOlympics is sponsored by**

Robotics Society of America

COMBOTS

SERVO MAGAZINE

# R/C Servo Controller

## Timer Based or Microcontroller Based

by Rob Caruso

**If you enjoy building things that move and have an interest in electronics, then there is a pretty good chance that you will end up using an R/C servo. We see them used frequently in robotics and definitely in R/C. And even though the servos used for robotics and R/C are the same, the controllers may not be.**

For radio control, a transmitter and receiver combination is typically used to control the servos. For most robotic applications, a specialized servo controller may be used and, in other applications, a multichannel controller connected to and commanded by a computer may be used.

The two controllers presented here — timer based and microcontroller based — are both single channel output which uses a potentiometer to set the servo's position. So, why then would we want to build a single channel servo controller when there are so many other types of servo controllers out there? Well, there are several answers to that question with the first being simplicity.

A single channel potentiometer controlled servo controller is a relatively easy device to design and build. The next answer is flexibility. With the controllers presented in this article, we can check a servo on a radio controlled vehicle at the field without the need for the transmitter — which also removes the need for the frequency. A single channel controller could also be used with a speed controller to break in or to test a motor. Also, because the device is self-contained, there is no need to boot up or occupy a computer to use it.

### Timer Based vs. Microcontroller Based

The first controller presented is based on the 555 timer. The second is based on a Microchip microcontroller. Why present two versions? First of all, each version has its pluses and minuses. For the timer based controller, no programming is required. For the microcontroller based unit, the parts count is less and it also has the capability to easily add more features. The timer version also offers a virtually infinite number of servo positions.

For the timer based controller, the tolerances of some of the components will have an effect on the timing of the output signal whereas component tolerances will not have an effect on the timing of the output signal in the microcontroller version. The current draw is less for the microcontroller version.

Presenting two versions also demonstrates that even though two devices may supply a similar output, the components used to generate the output may be very different. The style chosen to be built will most likely be based on personal preference as opposed to the pros or cons offered by each.

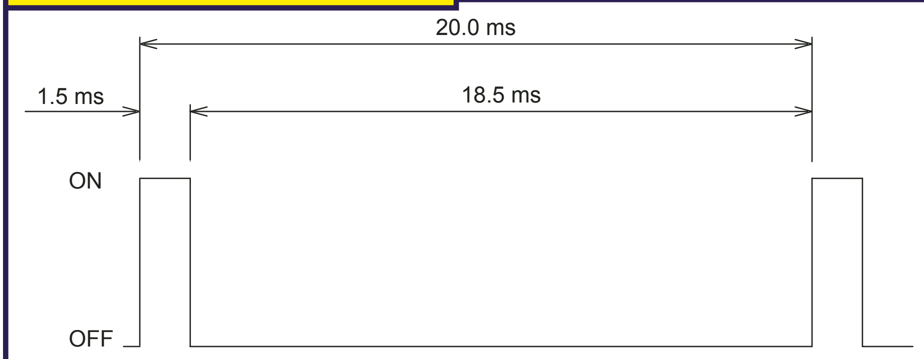
Whether you decide to build the timer version or the microcontroller version, there are several things common to both. The power supply is the same and both use a potentiometer for control. Most importantly though, the signal to control the servo's position is the same for both versions.

### Servo Control Signal

If we take a look at the control signal on an oscilloscope with the controller's potentiometer at its neutral or middle position, we see that the pulse width is approximately 1.5 ms and that its period is approximately 20 ms. (see Figure 1). The servos output shaft will also be at its neutral or middle position. If the potentiometer is rotated to its maximum counterclockwise position, the pulse width changes from 1.5 ms to 1.0 ms and the servos output shaft will rotate to one end of its travel.

Even though the pulse width changed, the signal's period did not. In other words, the start of each pulse is still 20 ms apart. Now if we rotate the potentiometer to its maximum clockwise position, the width of the pulse changes to 2.0 ms and the servo's output shaft will rotate to its other end of travel. Simply put, the width of the pulse is controlling the position of the servo's output shaft — shortening the

FIGURE 1. Servo Control Signal.



# R/C Servo Controller

pulse rotates the output shaft in one direction and increasing the pulse rotates the shaft in the opposite direction.

Varying a signal's on time is typically referred to as pulse width modulation (PWM). The servo's position is encoded into the control signal based on the amount of time that it is high. For a servo, the off time is not that important, as long as the servo gets a pulse about every 20 ms it will work without any problems. In order for us to control a servo, we need to simply recreate the signal. Also note that the control signal is only for controlling the servo's position; a separate lead is used to supply power to the servo.

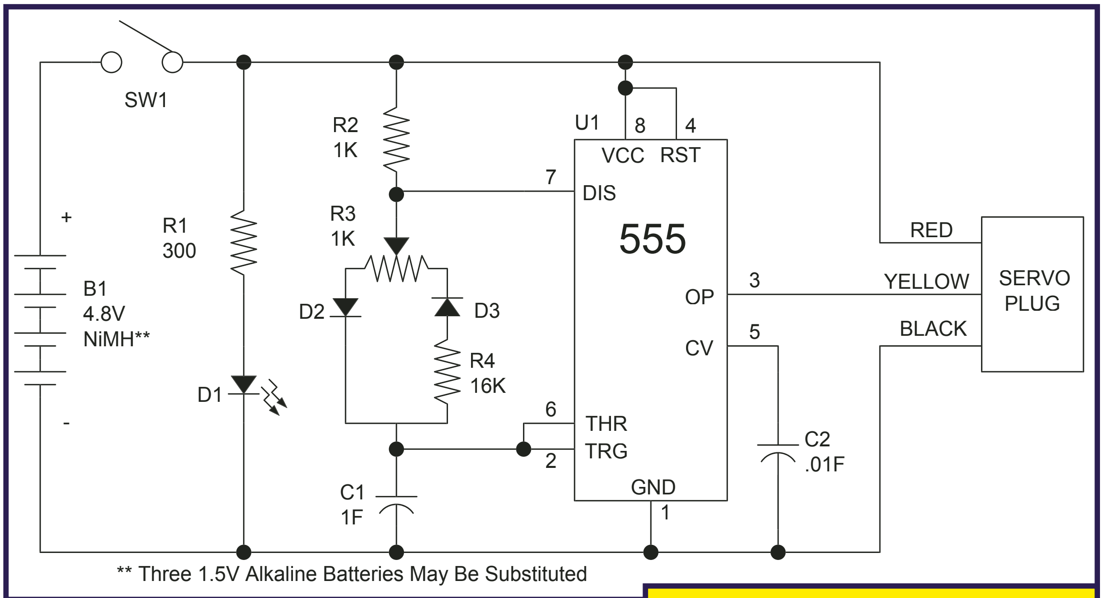



**FIGURE 2. Timer Based.**

## The Timer Based Version

As mentioned earlier, the timer version is based upon the 555 timer chip. Refer to Figure 2 for the schematic. The 555 timer chip was chosen because of its availability, output drive capability, ease of use, and low cost. In this application, it works quite well. The timer is set up as an astable or free running oscillator. The circuit is wired with separate charge and discharge paths for the capacitor. The reason that this is done is to allow us to vary the pulse's on time from 1 ms to 2 ms, while minimizing changes to the time between the start of each pulse.

In this application, it is not overly critical if the period varies, but our initial goal was to recreate the signal from an R/C receiver and this setup allows us to achieve that.

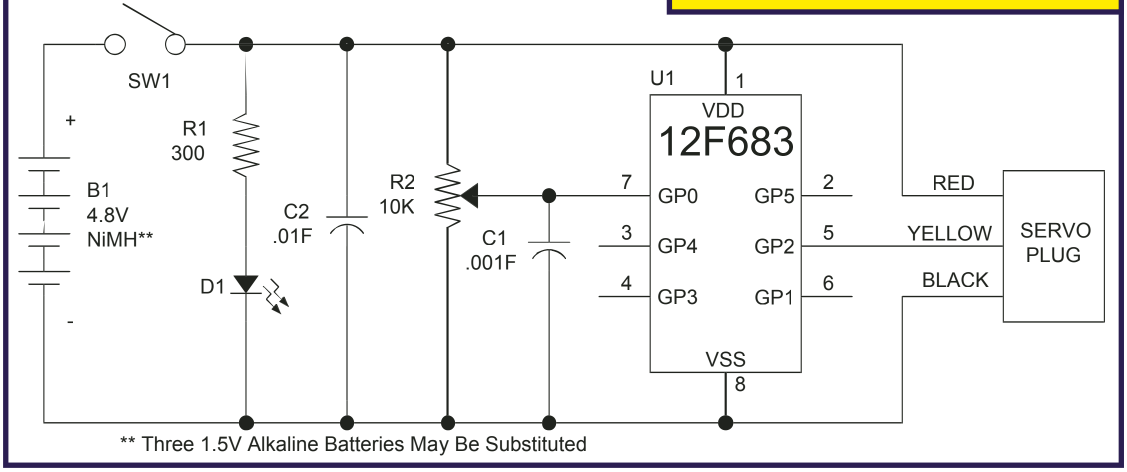
For the 555 timer, the output is on whenever the capacitor is charging. During a charge cycle, current flows through R2, then through the left half of R3, D2, and into C1. Capacitor C1 will continue to charge until it reaches two-thirds of Vcc. Once it reaches that point, a couple of things will happen.

One is that the output will turn off and the other is that the 555's internal discharge transistor will turn on. C1 will now start to discharge through R4, D3, the right half of R3, and into pin 7. The capacitor will discharge to one-third of Vcc where the output will turn on, the internal discharge transistor will turn off, and the cycle will repeat. Rotating R3 will change the amount of resistance in the charge and discharge paths. R3 is wired so that any resistance added to the charge path is removed from the discharge path and vice versa.

## The Microcontroller Based Version

The microcontroller version is based upon Microchip's PIC12F683 microcontroller (see Figure 3). The reasons for choosing this particular microcontroller are numerous. The 12F683 has a maxi-

**FIGURE 3. Microcontroller Based.**



# R/C Servo Controller

mum of six I/O and we only need two. We need one analog input to read the position of the potentiometer and one digital output to supply the control signal to the servo. The 12F683 has an analog-to-digital converter built into it. It also has an internal oscillator capable of providing several oscillator frequencies from 31 kHz to 8 MHz which is very helpful as we shall soon see.

The 12F683 has a built-in PWM which makes the programming a lot simpler. In order to use the internal PWM, we will need to set it up first. One of the things that we need to do is to set the period of the pulse width modulator. Looking back at our initial requirements, we determined that we would need a period of 20 ms. From the 12F683 datasheet, we see that to calculate the period's time, the formula is "Period = (Period Register #2 + 1) x 4 x Tosc x Tmr2 Prescale value." With an oscillator frequency of 250 kHz, we can get a period of 20 ms. To get the 20 ms period, we need to set the period 2 register to 78 (20 ms = (78 + 1) x 4 x 4 µS x 16).

Now that we have determined the period, we need to set the time that the PWM output is on. The on time is determined by a 10 bit value which is loaded into two registers. The eight most significant bits are located in the CCPR1L register and the two least significant bits are located in the CCP1CON register.

With 10 bits of resolution and a clock frequency of 250 kHz, we get a minimum step size of about 26 µS. If we do the math, we get 1 ms/26 µS which equals about 39 positions that we can set the servo to. With approximately 120 degrees of travel, that is about three degrees of movement per step. And even though

### Listing 1

```
//*******************************************************************//
// Servo Controller
// Microchip 12F683 Microcontroller
// 1mS to 2ms Pulse every 18mS
// Rob Caruso
// January 2006
//*******************************************************************//

#include <12F683.h>
#fuses INTRC_IO,NOWDT,NOPROTECT, NOMCLR, NOFCMEN, NOIESO, NOBROWNOUT, PUT
#use delay( clock = 4000000 )                       // SET INITIAL CLOCK TO 4MHz

void setup( void )                                  // SETUP THE A TO D, I/O AND CCP
{
    setup_adc_ports( sAN0 );                        // GP0 ANALOG INPUT
    setup_adc( ADC_CLOCK_DIV_8 );                   // ADC CLOCK
    set_adc_channel( 0 );                           // SELECT ADC CHANNEL 0
    setup_ccp1( ccp_pwm );                          // CONFIG CCP1 AS PWM
    setup_timer_2( T2_DIV_BY_16, 255, 1 );          // SETUP CLOCK AND PERIOD
    output_low(PIN_A1);                             // SET UNUSED I/O TO OUTPUT TURNED OFF
    output_low(PIN_A4);                             // SET UNUSED I/O TO OUTPUT TURNED OFF
    output_low(PIN_A5);                             // SET UNUSED I/O TO OUTPUT TURNED OFF
}

void main()
{                                                   // BEGINNING OF MAIN FUNCTION
int32 sum;                                          // DEFINE 32 BIT INTEGER VARAIABLE
int16 value, pwm;                                   // DEFINE 16 BIT INTEGER VARIABLES
int no_acqs, t2_value;                              // DEFINE INTEGER VARIABLE

    setup();                                        // SETUP A TO D and PWM
    while ( TRUE ) {                                // BEGINNING OF LOOP
        value = Read_ADC();                         // READ THE VALUE OF THE POTENTIOMETER
        sum = sum + value;                          // SUM THE VALUE
        no_acqs ++;                                 // INCREMENT THE NUMBER OF AQUISITIONS
        if ( no_acqs > 127 )                        // SEE IF WE HAVE ACQUIRED 128 READINGS
        {
                pwm = sum / 128;                    // AVERAGE THE POT READING
                set_pwm1_duty( pwm + 256 );         // SET THE PWM % + 1mS OFFSET
                sum = 0;                            // CLEAR THE SUM
                no_acqs = 0;                        // CLEAR THE ACQUISITION COUNTER
        }

        t2_value = get_timer2();                    // GET THE CURRENT VALUE OF TIMER #2
        if ( t2_value > 140 && t2_value < 205 )     // CHECK IF OSC SHOULD BE 250k or 4M
        {
                setup_oscillator( OSC_250KHZ );     // SET INTERNAL OSCILLATOR TO 250KHz
        }
        else
        {
                setup_oscillator( OSC_4MHZ );       // SET INTERNAL OSCILLATOR TO 4MHz
        }
    }                                               // END OF LOOP
}                                                   // END OF MAIN FUNCTION
```

# R/C Servo Controller

that might be okay for some applications, the servo will appear to be jerky or jumpy when it moves. But there is a way to get smoother positioning while maintaining a period close to 20 ms.

As mentioned earlier, the 12F683 has an internal oscillator with several settings ranging from 31 kHz to 8 MHz. The oscillator settings are in a register (OSCCON) which is accessible to the program running in the microcontroller. By making use of this feature, we can increase the clock frequency during the on time of the pulse to get better resolution for the pulse and then slow the clock down to get close to our desired period of 20 ms. If we redo the math with an oscillator frequency of 4 MHz, we get a minimum step size of about 4 μS which is short enough to smooth out the servo's motion and give us an adequate number of positions.

The program was written in C and compiled using CCS' PICM compiler (the hex file Servo_Controller.hex is available on the *SERVO* website, **www.servomagazine.com**). See Listing 1 for the C program. The program is grouped into four major sections: initialization of registers and I/O, reading and averaging the value of the potentiometer, loading the value into the PWM registers, and setting the internal oscillator frequency. So, where is the actual PWM timing done? The timing is done with the '683's internal hardware (see the 12F683 datasheet for details on exactly how it works). We just need to configure the PWM, set our desired on time, and it does the rest.

The program starts with a clock speed of 4 MHz and then reads timer #2 to determine if the clock speed should be switched to 250 kHz. We have to make sure that the oscillator is not switched to 250 kHz before the output pulse goes low; if we did, the width of the pulse would be stretched out and the servo would not function properly. Also note that the PWM output is internally routed to the GP2 pin.

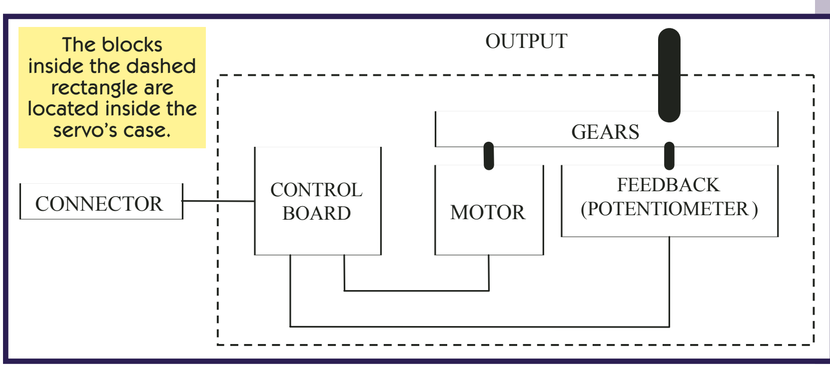
**FIGURE 4. Controller and Servo.**

## Construction and Assembly

Figure 4 shows the completed controller. My goal was to make the unit as small and as efficient as possible. To do that, I chose to use a small enclosure with a built-in nine-volt battery compartment and also to use an R/C receiver battery. The particular battery pack I used is a four-cell 1,800 mAH NiMH (nickel metal hydride) type. An 1,800 mAH cell has about the same capacity as an AA alkaline battery. And because the four NiMH batteries have a cell voltage of 1.2 V as opposed to the 1.5 V for alkaline batteries, we can eliminate the five-volt regulator circuitry typically used in circuits similar to this. (If you prefer to use alkaline batteries without a voltage regulator, be sure to use three as the maximum permissable voltage to the microcontroller is only 5.5 V.)

Most servos will work properly with a supply voltage anywhere in the range of 4.5 to 6.0 volts. I also chose to use connectors and ribbon cable to wire each device mounted to the case and connected to the main board. The power switch is mounted on the side of the case.

The circuit is built on a ONE PASSircuit OP840B prototyping board which was trimmed to fit. The micro-




### Anatomy of a Servo

A typical servo is composed of four major blocks: a control board, a motor, a series of gears, and a feedback device. The blocks are connected together to form a closed loop. The motor drives the servo's output shaft through the gears. The gears also couple the motor's shaft to the feedback device. The feedback device (usually a potentiometer) is used to determine the position of the output shaft and feed it back to the control board. The control board ties everything together. It receives the control signal through the connector and compares it to the feedback from the potentiometer.

If the potentiometer and output shaft are not at the position they are supposed to be, the control board determines which direction the motor needs to rotate and supplies voltage to the motor. As the potentiometer and output shaft get closer to the position they are supposed to be at, the voltage applied to the motor decreases. When the output shaft and potentiometer reach the position they are supposed to be at, the voltage supplied by the control board to the motor drops to zero and the motor stops moving.

The connector has three leads attached to it, one for the control signal and two for power. The power leads are typically red for plus and black for zero volts. The control signal lead can be yellow, white, blue, or orange.

# R/C Servo Controller

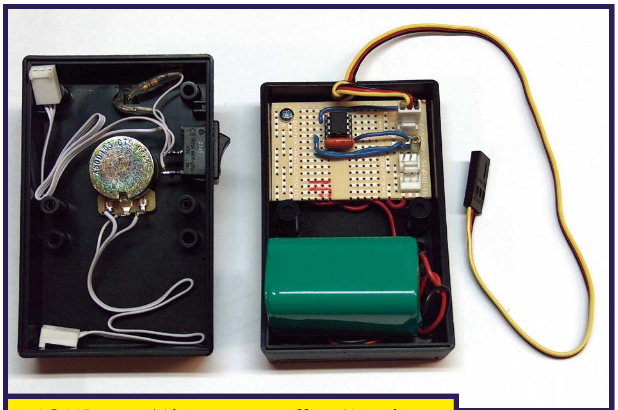


**FIGURE 5. Microcontroller Version.**

controller version is shown in Figure 5. The enclosure was also modified to accommodate the battery pack. The case must be separated to remove the battery pack but the battery cover allows access for charging the battery pack. A charging jack and SPDT power switch could also be wired in to simplify charging.

Because the circuits are fairly simple, several construction techniques including point-to-point wiring, wire wrap or a prototyping PCB could be used. A socket should be used for U1, regardless of how the circuit is assembled. I used a servo connector for a Hitec servo which came from a 12" servo extension lead. If you have servos with more than one style of connector, there are a couple things that you can do, including making an adaptor to convert from one style to another, wiring in multiple connectors, or using binding posts.

Testing the controller is straightforward. Check the voltage polarity on the socket pins without the IC in place. If the voltage polarity is correct, power down, install the U1 making sure it is inserted in its proper orientation, connect a servo, and power up. If you find that the servo is rotating in the wrong direction, then swap the two outer leads on R2.

As mentioned earlier, the component tolerances for the 555 timer version will have an effect on the output signal. If you find that you are not getting the range of control that you need, try a different potentiometer (R3) or capacitor (C1). The microcontroller version will not be affected by component tolerances.

## Ideas For Future Expansion

Both versions can easily be modified for servo reversing. A DPDT switch wired to swap the two outer leads of the potentiometer will do the trick.

The digital version could also be modified by replacing the potentiometer with a rotary encoder. The encoder could the feature of allowing to calibrate the dial position to the output shaft position — or mimicking a multi-turn potentiometer where several dial rotations are required to move from one end of travel to the other. If you're not the type that likes to rely on batteries, the unit could be converted for operation with a 12V car adapter.

I hope that you were able to get something out this article, whether it was how a servo works, how it is controlled, a new way to prototype, or a new way to use a microcontroller's internal clock. **SV**

### 12F683 Microcontroller Based Controller Parts List

- ■ B1 — 4.8V, four cell, 1,800 mAH NiMH battery or three 1.5V alkaline batteries
- ■ SW1 — SPST switch
- ■ R1 — 300Ω 1/4W resistor
- ■ D1 — Green LED
- ■ R2 — 10K potentiometer
- ■ C1 — .001 μF Capacitor
- ■ U1 — Microchip 12F683
- ■ C2 — .01 μF Capacitor
- ■ Miscellaneous: Case, prototyping board, knob, battery plug, eight pin DIP socket, female servo plug.
- ■ A programmed microcontroller is available for $9 USD.
- ■ US and Canada orders add $5 USD for shipping.
- ■ Inquire for shipping costs to all other countries.
- ■ Email orders and inquiries to caruso.rob@hotmail.com

### 555 Timer Based Controller Parts List

- ■ B1 — 4.8V, four cell, 1,800 mAH NiMH battery or three 1.5V alkaline batteries
- ■ SW1 — SPST switch
- ■ R1 — 300Ω 1/4W resistor
- ■ D1 — Green LED
- ■ R2 — 1K 1/4W resistor
- ■ R3 — 1K potentiometer
- ■ D2, D3 — 1N4148 diode
- ■ R4 — 16K resistor
- ■ C1 — 1 μF Capacitor
- ■ U1 — LM555 Timer
- ■ C2 — .01 μF Capacitor
- ■ Miscellaneous: Case, prototyping board, knob, battery plug, eight pin DIP socket, female servo plug.

### Sources

LM555 Timer Datasheet
*National Semiconductor*
**www.national.com**

12F683 μC Datasheet
*Microchip Corporation*
**www.microchip.com**

PIC C Compiler
*CCS, Inc.*
**www.ccsinfo.com**

Prototyping Board
*One Pas, Inc.*
**www.onepasinc.com**

Battery Connector and Female Servo Plug
*Available from most hobby shops*

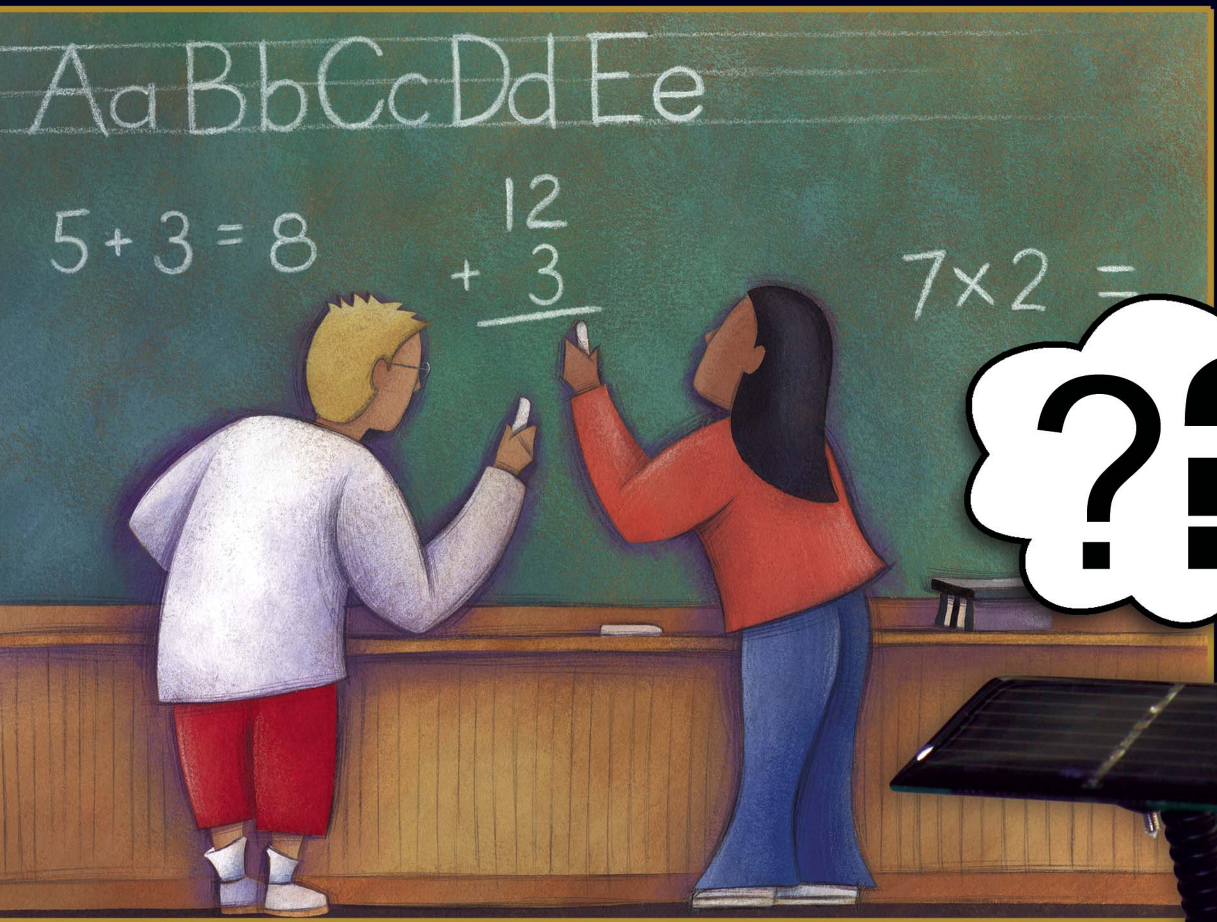
All Other Components
*Digikey*
**www.digikey.com**

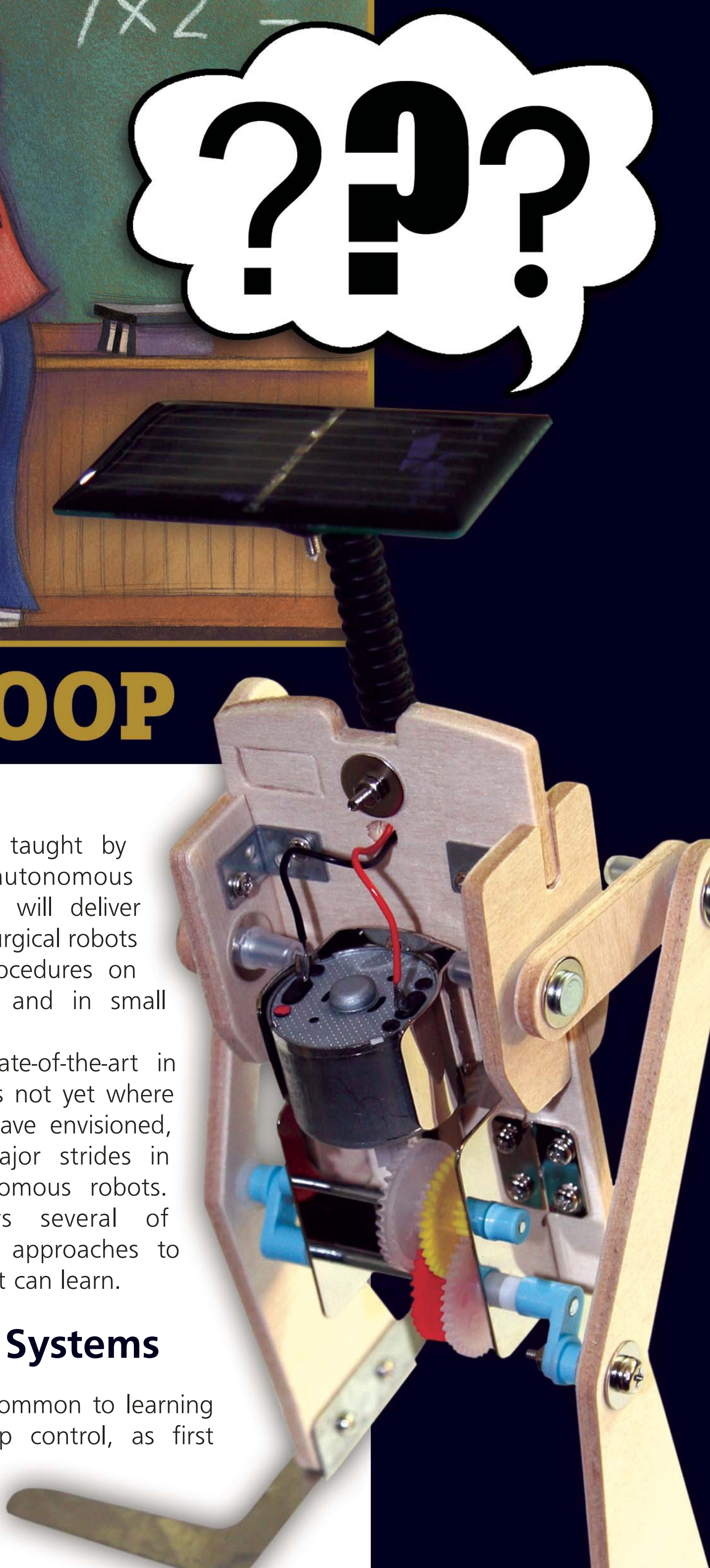
4.8V 1,800 mAH NiMH Receiver Battery
*All Batteries*
**www.all-batteries.com**

### About the Author

*Rob Caruso is an Industrial Automation Controls Designer. In his spare time, he enjoys spending time with his family and studying the art of Aikido. He can be reached at caruso.rob@hotmail.com*

# Robotic Learning




## CLOSING THE LOOP

— by Bryan Bergeron —

Developing intelligent, autonomous robots is one of the most challenging and exciting areas of robotics research. The requirements for a robot capable of operating in a novel environment with little or no external control are formidable: a robust physical structure and means of propulsion, an efficient energy management system, sensors appropriate to the environment and robot construction, effectors capable of interacting with the environment, internal and external communications capabilities, and a control system capable of adapting robot behavior to suit the environment.

Intelligence — the ability to learn from experience and adapt accordingly — is the gulf that separates contemporary hobby robots from future household service robots that can be taught by non-programmers, autonomous military vehicles that will deliver supplies in war, and surgical robots that will perform procedures on astronauts on Mars and in small community hospitals.

Although the state-of-the-art in robotics intelligence is not yet where Asimov and others have envisioned, there have been major strides in creating truly autonomous robots. This article reviews several of the most promising approaches to developing robots that can learn.

### Closed Loop Systems

A characteristic common to learning robots is closed loop control, as first

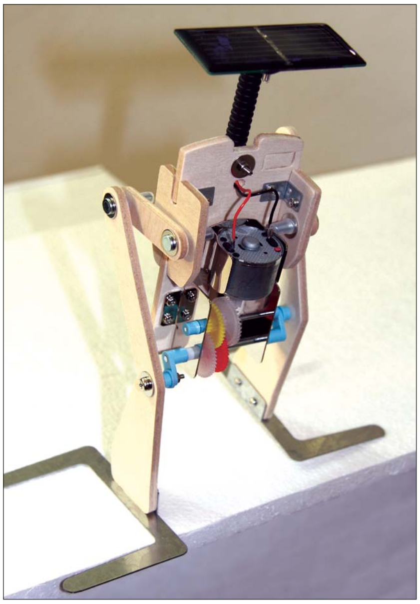


**Photo 1.** A solar robot about to plunge over the edge of a table, an extreme example of an open-loop system.

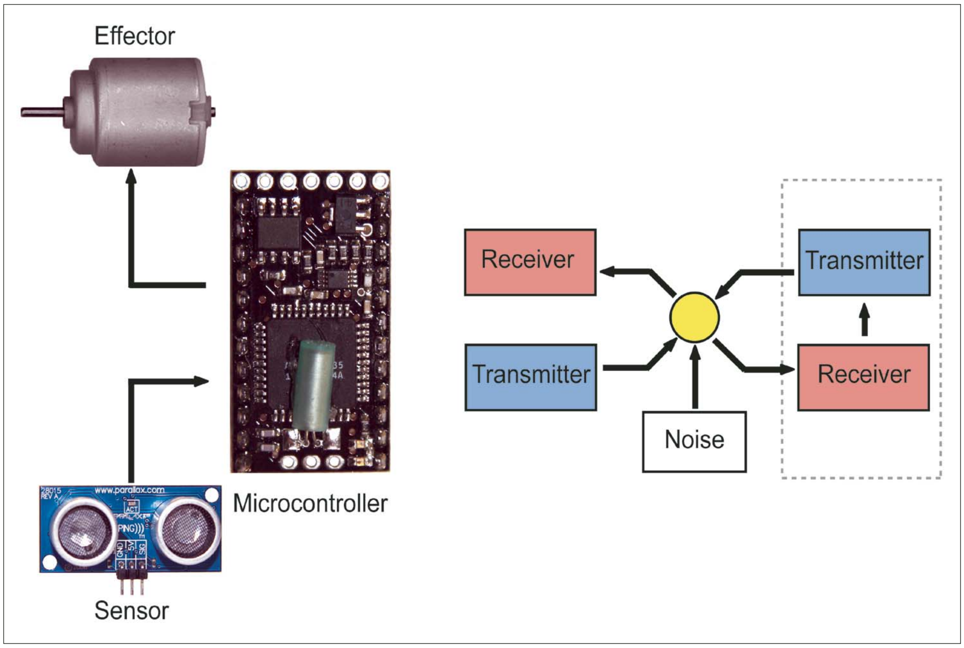



**Figure 1.** Closed loop robot architecture in which the microcontroller closes the loop.

described by Norbert Weiner in *Cybernetics*. To illustrate the importance of a closed loop or feedback in robot learning, consider the open loop robot in Photo 1. This solar robot — which consists of a solar panel, DC motor, and mechanical drive system — will walk off the table top until damaged beyond repair. The robot has no sensors to detect the table edge, no digital processor or analog circuit to respond to sensor signals, and no means of storing the experience of tumbling over the edge for future reference.

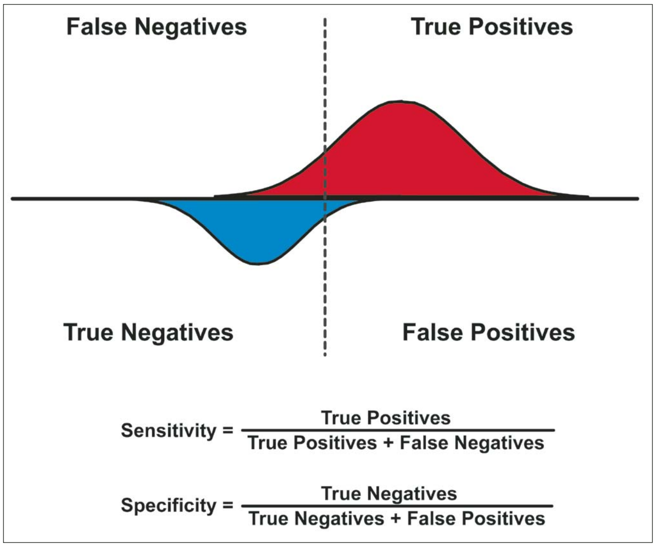
Designing and building simple solar robots has merit. However, from the perspective of robot learning, the evolutionary branch of robotics initiated by W. Grey Walter and his turtles is more relevant. Walter's turtles, like most hobby and industrial robots, employ some form of sensor-effector feedback.

As illustrated in Figure 1, regardless of whether the robot walks, crawls, rolls, or slithers, the typical configuration of a modern digital robot includes some type of sensor, effector, and controller. Signals from the sensor are transmitted to the I/O port of the controller, where they are processed. The controller closes the loop, in that the signals transmitted to the effector are a function of the signals received from the sensors.

Real-world robots are a mix of open- and closed-loop systems, by virtue of their physical design, type and arrangement of sensors, and conditions in the environment. Assuming the microcontroller shown in Figure 1 is programmed correctly, the robot should be able to avoid running into objects.

However, if the robot is designed without feelers or other means of testing the continuity of the surface, it could easily run off of a table top and plunge to the floor. Similarly, simply installing a microswitch and spring-steel whisker won't change the robot's behavior. The closed loop design is necessary but insufficient for the robot to avoid the perils of a table edge. New robot behavior has to be either programmed or learned.

**Figure 2.** Sensor sensitivity and specificity. The vertical bar represents the criterion for accepting and rejecting signals from the sensor.




## Noise

The architecture illustrated in Figure 1 has limitations that negatively impact robot learning. Following Shannon's Information Theory, the sensor and microcontroller are information sources and sinks that operate over an imperfect data channel. Variations in the environment, the nonlinearities, randomness and other imperfections in the sensor and effector, crosstalk, power supply fluctuations, errors introduced by the microcontroller's data acquisition and handling circuitry, and

other sources of variability can be modeled as a single noise source. Randomness, which is inherent in every sensor, is typically described in terms of accuracy, resolution (precision), repeatability, stability, sensitivity, and specificity.

One way of assessing the performance of a robot's sensor system is to determine sensor sensitivity and specificity. Given a criterion for when to call a signal significant, sensitivity is the percentage of actual positives that are counted as positive, whereas specificity is the percentage of actual negatives that are rejected.

Expressed another way, sensitivity is the number of true positives divided by the sum of true positives and false negatives, as illustrated in Figure 2. Similarly, specificity is the number of true negatives divided by the sum of false positives and true negatives.

Regardless of whether the source is mechanical or electronic, noise has a direct impact on the ability of a robot to respond to the environment and, by extension, to learn. Furthermore, the effect is cumulative, in that errors introduced by noise in the early stages of the robot sensor system propagate to and are amplified by later stages. As discussed below, the resulting uncertainty must be dealt with by the learning algorithm.

## Learning Robots

Biology is a convenient source of inspiration for robotic learning methodologies. The most promising approaches to creating intelligent, learning robots include simple reinforcement, complex reinforcement in the form of artificial neural networks and genetic algorithms, and a variety of statistical methods. Furthermore, biological systems suggest that learning requires memory.

In individuals, this memory typically takes the form of reorganized brain cell structures — an approach loosely approximated through the application of artificial neural networks. Species learning, in which instinctive behaviors are passed from one generation to the next, is commonly modeled by genetic algorithms.

While biological systems are often held as the gold standard for autonomous robot behavior, it's important to recognize the capabilities of even the simplest organic entities are beyond those of most robots.

Viruses — which aren't even living creatures — use the DNA, protein, and energy in host cells to replicate and mutate and adapt to changes in the environment. A single-cell amoeba is several orders of magnitude greater in complexity than a virus. The lowly amoeba can communicate and coordinate activity with other amoebas, detect enemies, replicate, travel under its own power, retreat from noxious stimuli, find and consume sources of energy, dispose of waste, navigate based on light, temperature, and chemicals.

Moreover, the amoeba can change behavior to suit the environment, and learn to avoid noxious stimuli. The amoeba can accomplish these feats in part because of numerous receptors, biochemical networks, and control mechanisms coded by a DNA library of 67 x $10^{10}$ base pairs.

Life at the microscopic and sub-microscopic levels is replete with models of autonomous learning behaviors. Even so, most developments in robotic learning parallel human learning. Model-based robotic learning, which approximates cognitive learning theory, is concerned with developing an internal representation or model of the environment.

In contrast, behaviorists focus observable interactions between robots and their environment. While there have been significant successes in each camp, contemporary robot researchers often integrate low-level behavior-based techniques with high-level model-based learning.

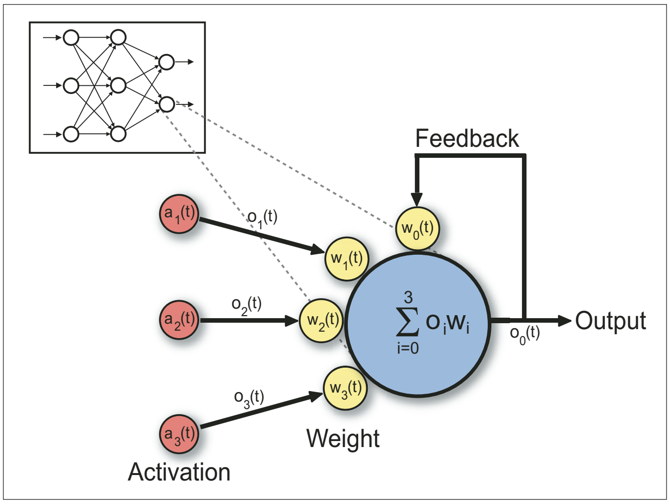



**Figure 3.** Artificial neural network showing details of the closed loop mechanism of an individual artificial neuron.

## Reinforcement Learning

Simple reinforcement learning involves assigning positive value to desirable behavior and/or negative value to undesirable behavior. For example, with a closed loop system, a robot can accrue positive scores by avoiding the edges of a tabletop, and negative scores by crashing to the floor. The efficiency and effectiveness of simple reinforcement learning depends on the complexity of the task to be learned and how the learned behavior is represented.

Two extensions of reinforcement learning that have been used extensively in robotics learning research are based on artificial neural networks and genetic algorithms.

Connectionists — those who study artificial neural networks — equate learning with modification of the synaptic weights of an artificial neural network. Artificial neural networks learn by extracting characteristics of the environment and forming weighted connections based on these characteristics.

As in biological systems, the sequence of training patterns determines the effectiveness and efficiency of learning. For example, artificial neural networks typically learn best if trained with simple cases and then progress to more complex cases.

An advantage of using artificial neural networks over simpler reinforce-

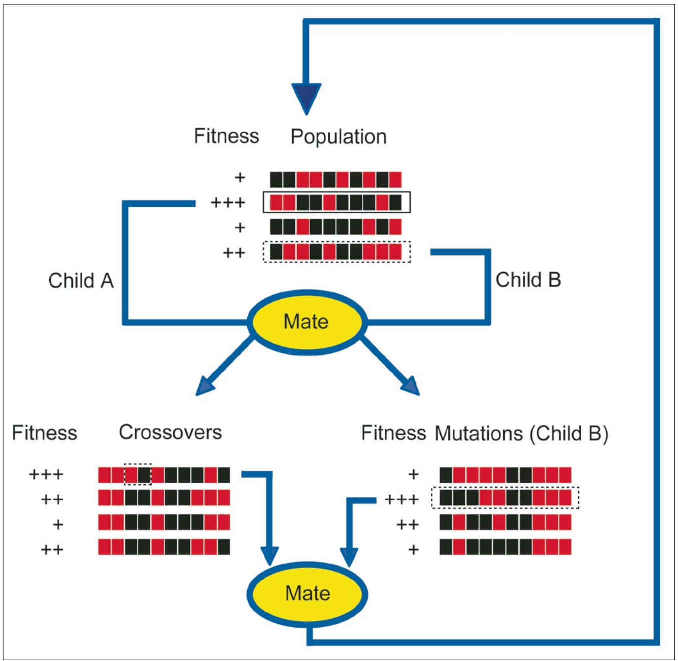



**Figure 4.** Data flow in a genetic algorithm showing the closed loop of crossovers and mutations followed by assessment of fitness to a user-defined function.

ment learning methods is that the design automatically deals with noise and uncertainty. Another benefit of using artificial neural networks as the basis of a learning robot is that learning can occur in near real-time using computationally tenable back-propagation.

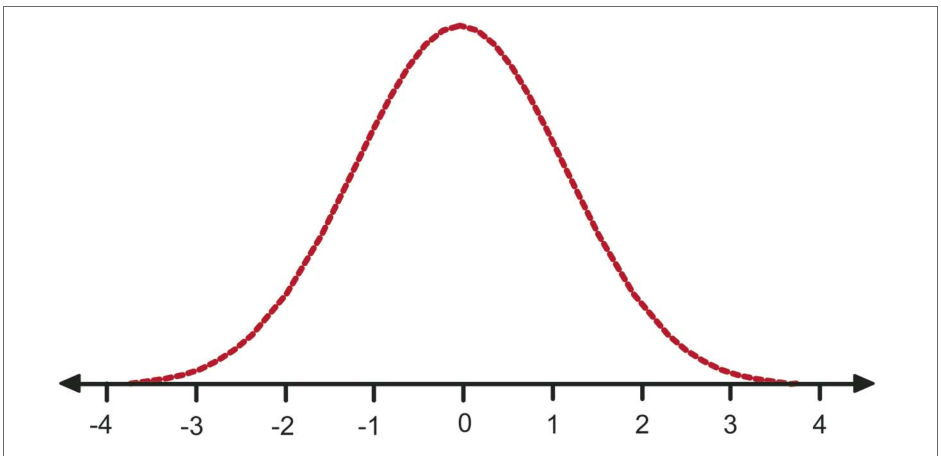
Unfortunately, the learning capabilities of artificial neural networks are a poor approximation of learning in real neural networks, leading researchers to supplement the technology with other techniques or look to other methods of robot learning. Neurons in real neural networks may have thousands of connections with other neurons.

Moreover, memory is encoded in structure, not simply the connection strength between nodes. In addition, there are numerous chemical modulators of biological neural activity that change the rate and effectiveness of learning as the chemicals wash over neurons and their synapses.

Genetic algorithms are a computationally intensive form of reinforcement learning that seeks to recursively identify the most fit learning robots, based on a fitness function. Unlike simple reinforcement learning that randomly seeks to satisfy a user-defined function based on positive and negative behaviors, genetic algorithms approximate the intermingling of chromosomal material in sexual reproduction.

As shown in Figure 4, robots in the population (or, more often, simulations of robots) are assigned fitness scores based on behaviors in the environment. Robots with the highest fitness functions are mated and mixed with crossovers and mutations, and the fittest robots are allowed to mate.

**Figure 5.** Belief distribution for sensor data as a probability density function.




Genetic algorithms have been used extensively in evolutionary robotics to create new behaviors, and sometimes these behaviors are more effective than those crafted by hand. Limitations of the approach to learning are linked to the computational requirements of genetic algorithms. Because considerably more processing power than is available on typical microcontrollers, batch, offline processing is the norm, often supplemented with robot simulations.

It's important to note that the various forms of reinforcement learning aren't mutually exclusive. Many investigators have developed learning robots that employ a hybrid approach based on artificial neural network and genetic algorithms. For example, a genetic algorithm can be used to define the initial weights in a neural network.

## Statistical Learning

Statistical algorithms, such as Bayesian Filters, can be used to create learning robots and as the basis for robots that can make decisions under conditions of uncertainty. Statistical methods, primarily probabilistic learning, rely on probability distributions to represent ambiguity and certainty. They can be used to estimate the environment based on data that can be inferred from imperfect sensor data.

Figure 5 shows sensor information displayed a probability density function, which represents the robot's internal model of the state of the environment. The robot can't be certain (probability <1) that sensor data reveals an obstacle, for example, but only infer the state of the environment.

Probabilistic algorithms have received a great deal of attention by the robotics community because they can be used to anticipate future uncertainty and consider this uncertainty in learning and decision making. A limitation of a probabilistic approach to learning is computational overhead. Bayes filters — the basis of most probabilistic learning methods — are computationally intractable.

As a compromise, approximations for Bayes filters are often used, including the popular Extended Kalman Filter.

These approximations introduce additional errors into the system. Another significant limitation of Bayesian methods is the assumption of independence of observations — an assumption that may be difficult to verify in practice.

Although the logic notation of Bayes formula can be daunting, the notation for Bayesian networks is a simple, compact graph. Nodes represent random variables and links represent the causal relationship between variables, as in Figure 6. The strength of the links is defined in terms of probability, and each node is associated with a conditional probability table that gives the probability of any outcome.

An alternative notation is to draw the probability tables in place of the circular nodes. Regardless of the notation used, the goal is to completely represent the logic captured by the learning system. In this regard, Bayesian learning methods are more transparent than artificial neural networks, in that the weights ascribed to the connections are often meaningless without extensive scrutiny.

## Future Directions

The most promising results in robot learning employ a hybrid approach that relies on internal models of the environment and sensor measurements. Successful systems also continue to use successful biological systems as a source of inspiration, from DNA and RNA, self-replicating inanimate systems (viruses) and bacteria, to animals visible to the naked eye.

Despite continued, measured success in robot learning, several issues have to be addressed before a surgical robot can learn by simply observing a master surgeon. Distinctions of offline or online learning through batch or incremental processing will dissolve as the next generation of multi-core processors becomes affordable and readily available.

More challenging are issues such as whether to use supervised or unsupervised learning. Unsupervised learning may provide new mechanisms, but the approach requires data of sufficient quantity and quality for learning algorithms to identify and extract meaningful patterns.

Working with learning robots will also present additional challenges to developers and end users. In addition to the structural overhead of increased complexity, learning robots incur additional costs, such as suboptimal behavior during the learning process.

Furthermore, this behavior may persist after learning if the interaction with the environment isn't representative of what will be encountered later. Most robotics enthusiasts look forward to these high-level challenges, because it will mean that robots capable of autonomous learning have finally arrived. SV

## REFERENCES


***Autonomous Robots: From Biological Inspiration to Implementation and Control,*** **by George Beckey. MIT Press, 2005.**

***Probabilistic Robotics,*** **by Sebastian Thrun, Wolfram Burgard, and Dieter Fox. MIT Press, 2005.**

***Cybernetics,*** **by Norbert Weiner. MIT Press, 1948.**

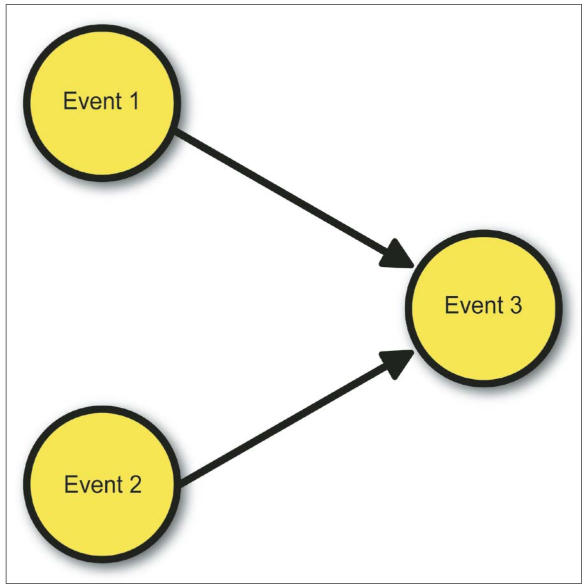




**Figure 6.** Bayesian network notation.

## ABOUT THE AUTHOR

*Bryan teaches in the HST Division of Harvard Medical School and MIT and develops AI-based applications and microprocessor-based augmented reality devices for the military and industry. You can reach him via email at BergeronBryan@Yahoo.Com*

# ROBOTS Who Live With People

by Tom Carroll

That's a pretty strange title, isn't it? In *SERVO*, I've written about promotional robots, industrial robots, space robots, undersea robots, and war robots. These robots are interesting to all of us, but we never get very close to them except through articles or TV shows. The one thing we all have is this secret desire to have a robot in our homes.

Back in the March '06 issue, I wrote about movie robots in my column, "Then and Now." It has been the movies that have inspired so many of us to delve deeper into the field of experimental robotics. Certainly, we all knew that these machines were either remotely-controlled or images generated by a computer.

Still, we really didn't want our bubble popped; we wanted to believe that robots that interacted with people were real. We not only wanted a robot who could act as a servant and could bring us a drink during the big game, but who could act as a friend and companion. We're not talking about an unnatural attraction to a machine here, but having a mechanical companion with whom we could communicate verbally to accomplish certain tasks.

## Early Thoughts on Artificial Intelligence

Did you notice the word "who" that I used in the title? Back in 1979, technical author and novelist, Pamela McCorduck wrote a "delicious book" (the words of *Scientific American*) entitled *Machines Who Think*. *Who* refers to an entity that uses computer-intelligent life. In the book, she introduced the program — Eliza — to the slowly maturing computer community, a small but growing group of people who were weaning themselves from the large mainframes and learning about microcomputers.

Eliza is a very interesting program that allows a computer to answer typed-in questions in much the same manner as a psychologist. It certainly was not developed to replace psychologists, but rather, was an exercise in computer intelligence or artificial intelligence.

I remember having the program on one of our lab's mini computers and asking it all sorts of questions when I should have been using the computer for serious tasks. The program would focus on certain names that you typed in such as "mother," "brother," etc., and would jokingly try to convince you that these people were the cause of your supposed "problems."

When I wasn't pounding the keys to input tricky questions, I was wondering just how such a program might work in a robot. Eliza became especially chilling when we hooked up a speech synthesizer and heard the answers returned to us in robot-like speech. It's rather interesting that synthesized speech came before speech recognition — just the opposite from Issac Asimov's *I-Robot* series. The robot — Robbie — in the first short story could respond to verbal commands, but only later models could actually speak back to their masters. This sort of interaction is what we all dreamed about three decades ago.

In May '04, I wrote an article in *SERVO* about promotional robots and some of the questionable tactics that people used to advertise these machines. In 1976, news articles were seen in papers, magazines, and TV touting a new company — Quasar Industries — that developed a $4,000 domestic android robot that could "do everything a maid could do and more, including vacuuming your rugs, serving you dinner, baby-sitting, and teaching your kids French."

A little "snake oil" sales talk went a long ways and robotics researchers soon proved the whole thing a hoax. Despite this negative publicity, it soon became evident that the public wanted robots in their own homes.

## The Robot of the 21st Century

The 21st Century is upon us and we are looking forward to leaps and bounds in advancements of robot technology. Author Daniel Ichbiah answers the question, "What exactly is a 21st century robot?" in his 2005 book, *Robots, From Science Fiction to Technological Revolution*. "It is a very powerful computer," he says, "with equally powerful software housed in a mobile body and able to act rationally on its perception of the world around it." He later speaks of evolution to a thinking android, which is a robot in the form of a man or woman — the final form that many wish for their mechanical servant. This android form was always the robot form of movies until non-android Huey, Dewey, and Louie of *Silent Running* fame and R2D2 of *Star Wars* entered the scene.

Let's break down his description and try to decide just what a robot is. Daniel mentions "It is a very powerful computer ..." What is that? Is it a Cray supercomputer, a mainframe, a mini workstation, or a Pentium 4 machine operating at 3.2 GHz with a 200 Gig hard drive and a Gig of RAM? Or, could it be a BASIC Stamp 2 driving a small desktop robot about a maze? What is "powerful software?" Is that 100,000 lines of high-level language code that can communicate with its human programmer or just 50 lines of machine code that can communicate efficiently with a handful of sensors?

Let's look at the "mobile body" part of the description. Must a true robot have mobile characteristics? Was HAL9000 in *2001, A Space Odyssey* not a robot because he was mounted in a large cabinet? Are the robots in factories not real robots because they are mounted on a floor? How about the part "... and able to act rationally on its perception of the world around it." Must the real robot be able to act rationally, not just react to stimuli around it? Hmmm. Must it be able to perceive the *whole* world around it, not just its immediate environment? Now that I've completely shredded

Daniel's robot definition, I must say that his definition is as good as any description around. The bottom line is that everyone really has his or her own definition, depending on what they want out of a robot.

## The Road to a Real Personal Assistant Robot

Since I've mentioned that everyone is different in their needs for a home robot, I'll trace out just a few of the robots that have been intended for home use over the years. Most of the robots that have been introduced were really intended for experimenter and hobbyist's uses. The Androbot TOPO was advertised as "the latest and greatest in the year 1, A.B" when it made its public debut at the January 1983 Consumer Electronics Show in Las Vegas, NV. This two-wheeled, blow-molded, three-foot high robot had no arms and wobbled like a kid's toy.

TOPO could really do nothing in the way of service for an owner. The company went bankrupt soon after. The next year, Tomy Corporation came out with several robots called Omnibots. The top of the line was a two-foot tall model called the Omnibot 2000 that had two arms — one which could serve drinks from a motorized tray. Another version could be verbally controlled, but these robots still did not act as true servants, either because of their small size or lack of functions.

In this same time period, a small company in California — Hubotics — brought out a 44-inch tall robot that was little more than a roving computer that could talk. With a monitor for a head and no arms, Hubot soon rolled into the sunset. In 1985, the Gemini robot from Arctec Systems was introduced. This robot was larger than all the others, but was still quite limited in its capabilities as a true home servant.

All these machines could roam about a home and converse with the occupants, but they still could do little useful *physical* tasks. In short, they were cute and great toys for those people who had money to throw away, but it had extremely limited usefulness as a home robot.

In talking with many of the developers of these machines over the past 20 years, these people all thought their machine was "the ultimate home appliance and would be in homes for years to come." It was "so much better than all the others." The times — they

are-a changin' — and robot technology is "light years" past the crude robots of two decades ago. Japan has taken the lead, but Figure 1 shows a trio of GeckoSystems Carebots — an American company that's competing with the many home robots being developed in Japan and Korea.

## What do People Want in a Home Robot?

About a dozen years ago, I was part of a panel that was tasked to determine just what people wanted in a home robot. Should it present a humanoid form and walk as a biped android like the Honda robot in development that we'd heard about? Should it be a quadruped pet such as the Sony Aibo prototypes that were in development? Should it have wheels or tank-type treads? Should it have arms and hands to handle items, and — if so — how much should these arms be capable of lifting?

Should the robot be voice-controlled? What sort of vision and sensor systems should it have? What sort of technology should be used to keep costs down, yet, have the capabilities and reliabilities that a consumer may want? These were — and are — tough questions that resulted in widely varied answers and opinions.

FIGURE 1. GeckoSystems CareBots.

## Robot Form

The first question that we tackled was a potential home robot's physical form. How should it look to its users? Dan Kara, of Robotics Trends and RoboNexus fame, presented an excellent argument for the humanoid form for robots in his *Appetizer* column in the January '06 issue of *SERVO* entitled "Why Humanoids?"

He presents his case by having the reader imagine what style of robot would draw their interest the most. Would 23 DARPA Grand Challenge autonomous vehicles racing over a 132-mile desert course draw more press attention than 10 humanoid robots racing on a 100-yard dash course?

For you and I as robotics experimenters and readers of *SERVO*, probably The DARPA Grand Challenge would draw our interest more, as we know the extreme complexities of computer, software, and sensor systems that these vehicles required.

However, Dan was right in picking the humanoid form for robots because he is referring to the general public who knows absolutely nothing about robot technology and can only go by what they see with their eyes and perceive as a real robot.

Small Roomba vacuum-cleaning robots and similar autonomous lawnmowers fit well with the general perception of robots, as they are basically single-function machines — not full-function servants. Our home robot panel pretty much decided that people who are not robot experimenters would feel most comfortable with some sort of humanoid form for a robot. We weren't talking about a C3P0 look-alike that walked and looked like a human in metallic "clothing," but a robot that was a bit shorter than a human with a similar-shaped body and "something" that a user could see was a head.

Since three of us on the panel were robot experimenters and the other three were in marketing and design, this vague body form was all that we could decide upon. We all did agree that a head was an important part of the robot's form, as it gave us humans a location upon which to focus our attention on the robot. Whether the head contained a vision system and a voice system was not important.

## Robot Locomotion

The next question tackled was the

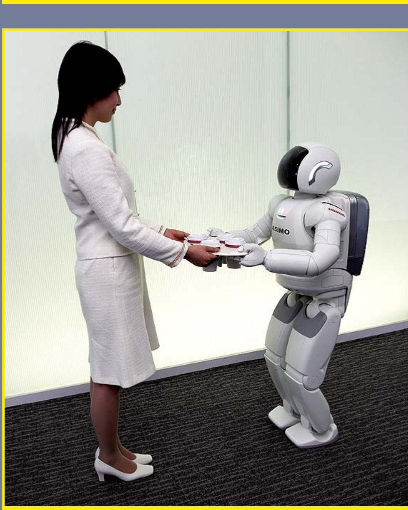


FIGURE 2. Honda's Asimo.

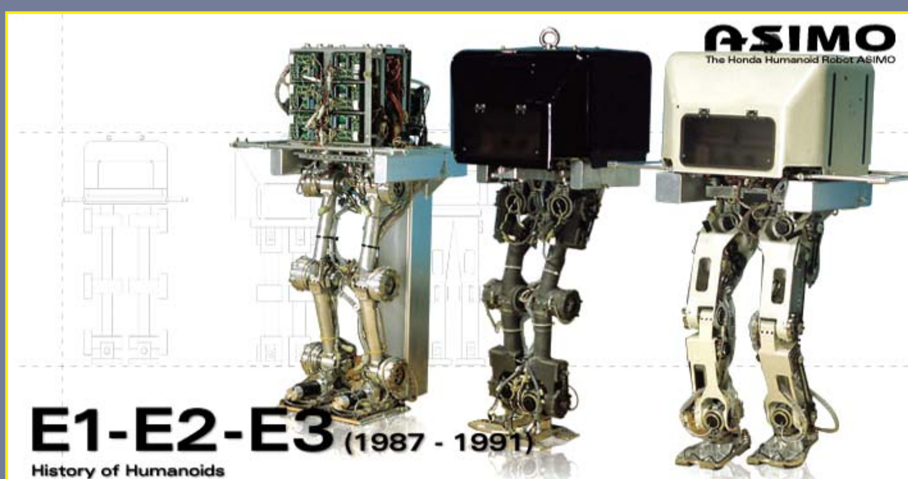



FIGURE 3. Honda's early E-series.

method of locomotion. Mobility is the one factor that distinguishes a "home robot" from its industrial cousins fastened to a factory floor. The robot must be able to come and go to a location as requested by its owner.

Bipeds are a natural and would seem most applicable for a robot in the home, but their inherent instability around people soon became apparent. Honda has been quite successful with the introduction of its robot — Asimo — shown in Figure 2.

This amazing bipedal robot's development did not come cheap or easily. Honda started with the goal of producing a bipedal walking robot and did not even stop to examine other motive methods. Walking technology research was begun long before the "E Series" shown in Figure 3 and the later "P series" shown in Figure 4.

The fact remains a small child or an elderly person could easily topple a bipedal robot, whether intentionally or accidentally, creating costly damage to the robot and possible injuries to a human. The robot cannot react fast enough to prevent a fall. Can you imagine the scenario of a robot calling "Help, I've fallen and can't get up" and its elderly owner coming to the robot's rescue? It's supposed to be the other way around, right?

A quadruped like Sony's Aibo cat/dog or a hexapod would be more stable than a biped, but would take up more floor space and you'd end up with a spider-style robot similar to the Odetics Odex — not very friendly looking.

Legs and walking have the distinct advantage of being able to traverse uneven surfaces and even walk up and down stairs, something that only very complex wheeled robot bases can accomplish. A derivative of Dean Kamen's amazing IBOT wheelchair — a six-wheel machine that goes up and down stairs and cruises effortlessly through sand or gravel — might prove to be the motive method for the robot of the home.

NASA has attached its Robonaut space robot to one of Kamen's Segway Transporters as a laboratory demonstrator (see Figures 5 and 6). Our group finally decided that some sort of wheeled base was far safer, more reliable, and less complex than legs, though we all were secretly hoping that walking technology would advance to the point in the near future of being the most functional.

## Robot Arms

Since our group was studying robots for the home, android or humanoid configurations were the first designs that we considered. People feel at home with something that looks somewhat like themselves, as *Star Wars'* C3P0 proved, though many more people seemed to adore little R2D2 — a robot that looked more like a beeping shop-vac than a functioning machine. Though we'd decided against legs, we were all adamant that the ideal home robot should have arms — functional arms. Now comes the tough part ... just what should these arms be capable of doing?

As we know, it is the robotic arm that encompasses all that an industrial robot really is. It is this arm that allows the lowly factory robot to earn its keep. Add a specific "end effector" for the particular task at hand, and you have a robot ready to be programmed for the job. There are more types of end effectors available for robots than robots to use them, so where does a designer begin? The home robot's end effector should be close to a human hand or a claw with an opposable thumb.

When it got down to the business of deciding on just what type of arm was best, everyone had his or her own idea. The typical "human" style arms that had the shoulder joint at the top of the robot's chest structure with a lower elbow connected to a forearm was what came to our minds first. This type of arm — a modified version of the industrial robot's "revolute configuration" — can easily reach down into a cavity and still reach up over the robot's head. The bad part about this type of arm is it uses a lot of energy to lift something.

Just think how hard it is to "curl" a 50-pound barbell or lift a 150-pound barbell over your head. With the curl, you have to exert over 50 foot pounds of torque at the elbow — a hard task for a small mechanical joint in a robot. This is why a 3,000-pound industrial robot

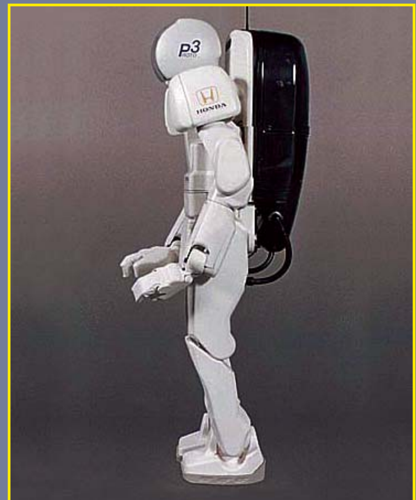


FIGURE 4. P-series.


FIGURE 5. NASA Robonaut on Segway.


FIGURE 6. I-bot.


FIGURE 7. Mitsubishi Wakamaru.

cannot lift nearly as much as a 175-pound person. The industrial robot can move a smaller mass much faster and with far greater accuracy, but these specifications are not required in a home robot. The Mitsubishi Wakamaru robot utilizes the revolute arm configuration with a humanoid body on a stable, wheeled base (see Figure 7).

I happen to be a proponent of the SCARA (Selective Compliance Assembly Robot Arm) robot arm configuration, a type of industrial robot that you see in circuit board assembly and similar light tasks, though some large SCARA arms can lift several hundred pound pallets. Refer to Figure 8.

Notice how the arm's "shoulder" can be moved up and down on the slotted extrusion; envision a leadscrew driving it up and down. SCARA arms have vertical axes, like the hinges on a door, so less force is required to move a large mass. This type of arm cannot reach down into a cavity or upward above the robot — a limitation for many applications. It can, with the use of a long leadscrew or ballscrew attached to the arm's shoulder, lift heavy objects quite easily, just as a person using a bumper-type car jack with a hand-cranked leadscrew can easily lift a car.

A personal assistant robot that I designed to physically assist seniors in independent living utilized two SCARA arms riding in 48-inch ball-screw driven tracks on the sides of the robot. I've highlighted just two of many styles of robot arms; each configuration will excel in specific applications.

## Robot Control

How the robot was controlled and functioned among people was difficult to "nail down." The ultimate "control" would be total autonomy, but that remains a future possibility for a complex personal assistant robot. Ultrasonic sonar arrays, active IR sensors, passive IR sensors, sophisticated vision, and speech cognition systems connected to a computer with well-developed software can do a lot to make today's robots fairly autonomous, but these have to be honed a bit more.

The average person would think that the latest $100,000 industrial robot would be the most advanced robot a person could buy compared to a university-developed home robot, but the opposite is true. An industrial robot operates in a fixed environment that usually does not change. It is programmed to know everything in its operating envelope, though a vision system may be employed to determine orientation of parts on a conveyor belt or similar.

A home robot operates in an "unfriendly" and changing environment as it roves about a house. An object that was on a table yesterday may be lying on the floor today, right in the way of the robot's intended path. A mischievous child may be intent on directing the robot out onto a busy street. Do you remember when Robin Williams — as the robot in *Bicentennial Man* — was told to jump out the upstairs window by one of the girls in the home? He did. Oops!

Honda's Asimo seems to be fully autonomous as he walks about a stage in front of crowds or ascends stairs in a convention center. What you don't see are the engineers out of sight with hands on remote controls. He used a full workstation for the P3 model and uses portable controls for Asimo. You can't see the beads of sweat dripping from their brows as he descends the stairs; their thoughts of "Oh boy, don't let him fall!"

Yes, he has very sophisticated walking gait software honed over many years, driving complex leg joints. There is also navigational, turning, and many other movements programmed into him, but a person is still acting as a supervisor off to the side. Wheeled robots can act autonomously in a crowd, mindlessly roaming about with the appearance of intelligence. But, can they act and react to perform a complex task? Can they save a child from being killed as in Issac Asimov's *I Robot*? Robbie saw a programmed tractor trundling down a path towards little Gloria and he ran out and scooped her up just in time.

Robot control varies from as much autonomy as we can develop to total remote control. Most developers decide on the "middle ground" of using some autonomy tied to some sort of remote control. That control can be by voice commands or by radio control. Feedback can be through direct observation or by an onboard video camera with images fed back to a remote control panel.

IBM has recently released the IBM Embedded ViaVoice 4.4 software package that includes freeform command recognition, a new voice recognition technology that can allow users to speak commands naturally without memorizing specific predetermined commands. I personally believe this will be great for robot control. I've always had a pet peeve about the term "voice recognition." My cat "recognizes" my voice and I can "recognize" Greek being spoken, but neither of us has any idea of what is being "said." How about using the term "speech understanding" or "voice cognition?"

> **"The one thing we all have is this secret desire to have a robot in our homes."**

## Robot Power

Honda's Asimo may be able to walk at a bit over 1.5 mph (2.7 km/h) and even run at 6 km/h, but to carry his 115 pound, 51 inch tall body, a 10 amp hour 38.4 NiMH battery is required, and that only lasts 40 minutes when he is walking. The battery is placed in his backpack — a good location for a walk-

ing bipedal robot — though some experimenters have actually placed weighty batteries in the foot areas and have actually improved the robot's walking.

It takes a lot of power to drive the motors for his 34 degrees of freedom. Most experimenters' large robots start out with gelled electrolyte batteries, and they sometimes progress to nickel metal hydride or similar chemistries. Sterling engines, solar power, and other methods can be used. Higher voltages, such as in Asimo, can result in smaller wiring being required, though most experimenters end up using five volts for the logic, 12 volts for most motors, and maybe 24 volts for the largest motors.

Efficient controllers, properly-selected PWM frequencies, efficient motors, and good mechanical design can all help to cut down power requirements and thus increase operational time or decrease weight. These are some of the main design criteria for home robot design, but there are so many more to consider, such as sensors, sensor placement, mass properties, aesthetics — the list goes on. This complexity is what makes robot design and development so interesting.

## The Ultimate Home Robot

Back in 1984, I wrote an article in *Popular Mechanics* entitled "The Smartest Homebuilt Robots," in which one of the featured robots was a robot of mine I called Ultima. Wow, what a pompous name. The little robot was only 28 inches tall and the only thing "ultimate" about it was the two off-road winch-powered arms that could out arm-wrestle King Kong. It's funny, though, no large apes were standing in line to test it.

It really could do nothing useful except eat money. The arm motors "ate" over 100 amps each and that was the reason I started looking at SCARA arms for my personal assistant robot project to allow seniors independent living. That project was very promising until my main venture capitalist had to back out during the dot.com crash.

Home robot development takes lots of money, yet, the near-term applications are very evident. Sony recently made the very difficult decision to halt production of their popular Aibo robots, as well as further development of the Qrio humanoid robot — a walker similar to Honda's Asimo. The reason was cost of development vs. return on investment — a basic business decision. Sony sold over 150,000 Aibos at around $2,000 each, but development costs were very high and only so many consumers could really afford them. US-based I Robot, on the other hand, has concentrated on home appliance robots with their Roomba and has sold over 1.5 million of them.

FIGURE 8. Bosch SCARA Robot.


FIGURE 9. HelpMate Robot.

Even with the demise of Sony's robot business, it looks as if the ultimate personal assistant home robot will still come to us from across the Pacific. Asian cultures have long seen the need for robotic help within the home and their corporations are spending hundreds of millions of dollars in research and development. Smaller US companies, such as GeckoSystems, are refining their products, but it will be the consumers and the largest companies who drive the market.

Joe Engelberger, the founder of Unimation and known as the father of Industrial Robotics, brought forth the Helpmate robot to work in hospitals among people, delivering meals and supplies to the staff (see Figure 9). It was an uphill battle all the way for him, but Joe has always wanted to build a great home robot. Maybe he will, but, as it stands, I do not have a clue as to who, or what, company will design and manufacture the first robot who will live with people. **SV**

by Dave Calkins

# Roboexotica
## The Festival for Cocktail Robotics

What are the three greatest things in life? In this month's episode of robot competitions around the world, we'll be dealing with the second and third items on that list — robots and cocktails (we'll cover sex and robots in next month's article.)

The Austrian groups "Monochrome" and "Shifz" co-sponsor what is probably the greatest robot competition in the world — Roboexotica — which deals solely with robots as bartenders and drinking companions. Robots shouldn't be gladiators, they should be your buddies! And Roboexotica brings that philosophy home (or at least it brings it to Vienna!). Just as cocktails themselves are international in their nature, so were the entrants. Robots came from around the world to show their mixological abilities.

I realize that there is a lot of charm to sitting on a bar stool and chatting with the bartender about how your girlfriend doesn't understand you, but sometimes people just want a drink delivered quickly — or they want to be amused while they're drinking ('cause you know, not every bartender or cocktail waiter/waitress is all that funny or endearing).

There are five different competitions at Roboexotica:

• *Mixing* — These robots are robots that make you a cocktail — some of them specialize in one or two drinks, while others can whip up over 200.

• *Serving* — Serving bots bring cocktails to patrons, regardless of who made the drink.

• *Conversation* — Companion bots keep you company while you slowly get sauced. It's not like anyone else wants to hear the same old story about how you saved the company thousands of dollars by re-writing that program in 1996 ...

• *Lighting Cigars/Cigarettes* — Since smoking is such a big part of the Austrian culture, people need cigarette-lighting bots that can whip out a Zippo faster than James Bond.

• *Special Achievement* — Those robots that perform multiple functions or are just exceptionally above the rest can go for the Special Achievement award (think "Best of Show").

"One reason for holding Roboexotica is to show the bright side of an automated present/future," said event creator Magnus Wurzer, of the Austrian group Shifz. "Another reason is to meet with people from different fields of occupation to discuss with them how we can fight the dark side of an inhumane environ."

"The mix of events is an adequately comfortable setting for real human interaction between creators, philosophers, artists, and scientists. As lately there is a rise in commercial approaches to automatic bartending, an important mission for Roboexotica is also to promote the values of grace and style in contrast to efficiency and profitability. Values that should be upheld in an electronic (cocktail) culture."

For example, mojitos have become very popular in the last few years — and with good reason. Good rum, sugar, and freshly mashed mint and limes — it makes for a very tasty and refreshing cocktail. But many bartenders loathe to make you one due to the time and effort involved in making them. Consider if you will then, the wonder that is Robomoji — a marvelous Rube Goldbergesque robot made by Austrian Robert Martin, that makes you a fresh mojito without any complaints but with a free serving of charm.

> **“The Austrian groups “Monochrome” and “Shifz” co-sponsor what is probably the greatest robot competition in the world — Roboexotica — which deals solely with robots as bartenders and drinking companions.”**

Photo 1

Photo 3

Photo 4

Photo 2

Photo 5

**PHOTO 1.** Alan by Chris Veigl is always keeping an eye on the customers.

**PHOTO 2.** Franz Ablinger makes some adjustments to his creation, Gesundheit4.

**PHOTO 3.** Kal Spelletich’s Whisky Pourer is simple and elegant, but gets the job done.

**PHOTO 4.** Chapek pours himself a martini, while a scary bearded guy watches the action.

**PHOTO 5.** Cockbot is not only efficient, but a work of art. It makes a good variety of cocktails amid a beautiful setting.

Upon paying, your glass moves out from a giant horizontal gear onto a chain drive (no ordinary chain here — this is a real one gauge hurricane chain!). Two clear mechatronic boxes hover above your glass at the first stage. The first clear reservoir is filled with fresh mint leaves, which are ground out and into your glass using a giant sharp-toothed sprocket inside the casing. A second identical reservoir slowly grinds fresh chopped ice into your glass.

The chain drive slowly moves your glass along to the next stage, where your glass awaits a delivery of fresh fruit. A fresh lime is released from an upper tank two feet above your glass, and slowly rolls down a twisting path, where it finally gets chopped in half before having all the juice squeezed out of it to join the mint and ice. The sounds of a giant saw tooth blade are all you hear as the chain slowly advances to the next stage, where a pneumatically-powered wooden mortar then pounds your lime, mint, and ice into an aromatic mix which perks up your appetite.

The final stage adds simple syrup (sugar water, held in a Rose's lime bottle) and spiced rum to the glass, filling it via an electrically actuated pouring system guaranteeing a perfect mix and eliminating short-pours. The saw blade continues its drudgery and your newly formed cocktail makes its way to the end of the robot and into your greedy little mitt. You got a great show, a perfectly mixed cocktail, no snotty comments, and you don't even have to tip. *It's the wave of the future I tell you!*

"Most of the 'cocktail robots' that were on exhibit through [previous] years were mostly put together by artists and electronic tinkerers," said Wurzer, "but Roboexotica has also tried to connect with industrial robot builders."

This year, industrial cocktail robots held court due to a team of university students from Fachhochschule Mechatronik in Vienna. Das Tier (the animal) is probably the most expensive drink mixing robot in history. Using a $100,000 industrial robot arm, the robot would make you either mulled wine or rum-spiked tea. Its oddly hypnotic arm whipped around at lightning speeds behind a wooden reindeer cutout, jumping from bottles of wine to mixer, before gently placing the Styrofoam cup onto a lazy susan for the lucky guest.

Cockbot and Gesundheit IV followed Das Tier's basic design, and mixed up fine martinis, but with far cheaper parts. Cockbot made excellent martinis, while Gesundheit IV could mix up either a screwdriver or a straight orange juice. All three were strictly industrial machines, though. Unlike Chapek, who was an anthropomorphic talking head and arm. Chapek would pour you a martini while happily insulting you. At least, he'd make drinks for customers when he wasn't kicking back the martinis himself. (What can I say, it was a flaw in the programming.)

For those who wanted to keep it simple, Kal Spelletich's Whiskey Pourer would set you up with a shot of booze quickly and neatly. No spilling and no snotty comments. And, of course, no need to tip.

All in all, Roboexotica was a fantastic event. I really wish that there were more art-bot events in the US. Of course, the cocktail robots can serve non-alcoholic drinks, but there's always a lot of fun to be had in making and displaying these robots. Certainly, the market for cocktail-making robots is huge. Drinking will always be around, and so will bars — but what bar manager wouldn't want a bartender who never called in sick, always poured out exact amounts, and didn't have nasty lovers' quarrels with the waitresses every other weekend?

## ROBOEXOTICA 2005 CHAMPIONS

**Mixing**
1. Robomoji — Robert Martin (Germany)

**Serving**
1. Das Tier — Fachhochschule Mechatronik (Austria)

**Special Achievement (tie)**
1. Chapek — David Calkins (USA)
2. BBQ — Kal Spelletich (USA)

**Conversation**
1. SMS-Bots — Sergei Teterin (Russia)

**Lighting Cigars/Cigarettes**
1. Bic-o-mat Plus — Adrian Dabrowski (Austria)

**Runners Up**
1. Alan — Chris Veigl
2. Bowlebot — nur schrec!
3. Cockbot 1 V 5.0 — Chris Veigl
4. Flamebot — nur schrec!
5. Gastone — Jannick Schmidt
6. Gesundheit IV — Franz Ablinger
7. RoboFriend — Thomas Heike
8. Servebot — Lukas Bramhas
9. Sicky — Mic Wlodokowski
10. Tender One — Redl Barsysteme
11. Whiskey Pourer — Kal Spelletich

## Cocktail Robot Events In 2006

If you're looking to build a cocktail robot, your options are sadly limited. The next Roboexotica will be held in Austria, December 5th-9th, 2006 (**www.roboexotica.org**). In the US, ROBOlympics (June 16th-18th in San Francisco, CA) will sponsor both cocktail mixing robot events, along with other art robot events, combat robots, sumo, the *SERVO*-sponsored 2006 Tetsujin event, and many other robotics competitions (**www.robolympics.net**)

Next month, I'll be covering the RFL National Championship and ComBots Cup — Combat Robotics isn't dead — it's alive and well, growing, and the lucky winners took home $10,000! SV

Photo 6

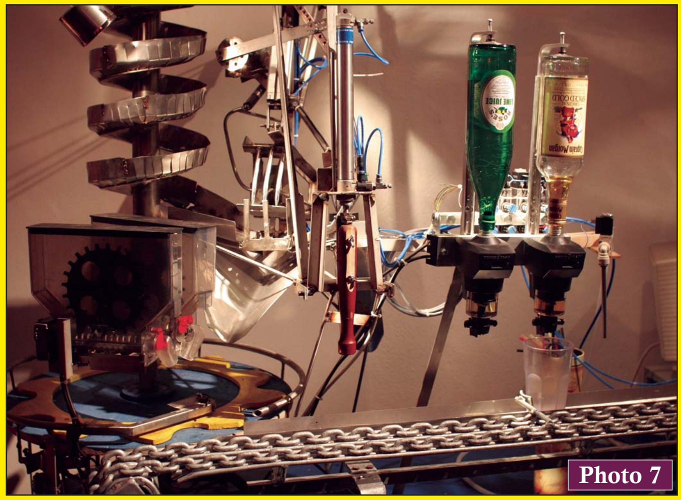


Photo 7


Photo 8

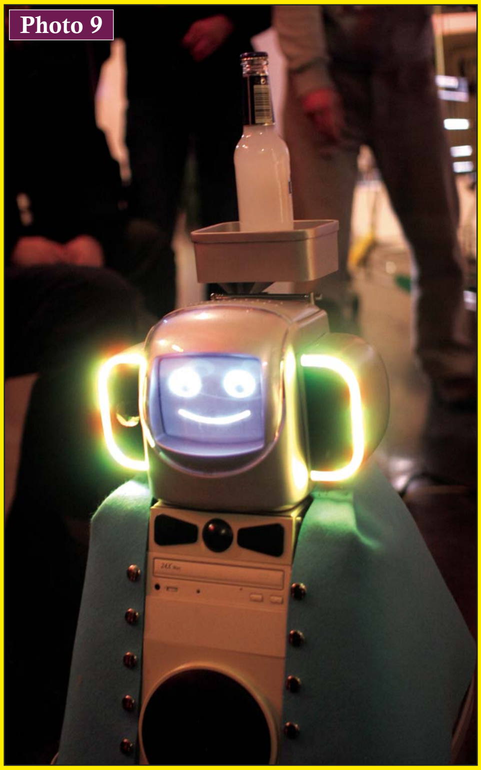


Photo 9

Photo 10

Photo 11

**PHOTO 6.** The front of Das Tier is a happy reindeer. Perhaps he's had too much of the Mulled Wine?

**PHOTO 7.** RoboMoji is primed and ready to mix you a cocktail.

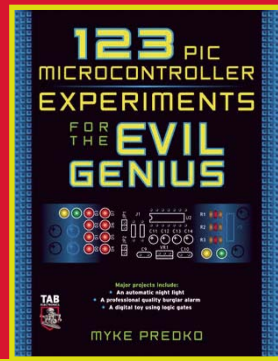
**PHOTO 8.** Gesundheit4 serves up a perfect glass every time.

**PHOTO 9.** Robofriend by Tom Heike will be happy to serve you!

**PHOTO 10.** The Roboexotica logo is not only drinking a tasty fruity beverage, but is fully wired for serial input and has many hidden robot parts (if you're willing to look.)

**PHOTO 11.** All those drinks wouldn't be nearly as good without some tasty barbequed snacks. Just be careful that the jaws from the BBQbot don't get you before you get your meals.

# 123 PIC MICROCONTROLLER EXPERIMENTS FOR THE EVIL GENIUS




## Book Review

by Matthew Nuzum

Have you ever wanted to create your own robot or build your own electronic device that interfaces with a PC? Or how about building a new high-tech toy that will blow the socks off of your friends? Well, I have, and recently I spent some birthday money on an excellent book titled *123 PIC Microcontroller Experiments for the Evil Genius* by Myke Predko

This is not Myke Predko's first book on the subject. An earlier book — *Programming and Customizing PICmicro Microcontrollers* — is one of the must-have books for serious PICmicro developers. *123 PIC Microcontroller Experiments* caters more to technical enthusiasts who are interested in getting started with the platform.

I am really pleased with the quality of this book. As many have found out, developing with the immensely popular PIC microcontroller (MCU) involves learning the hardware, learning the programming, and maybe even learning electronics.

This can be quite a barrier to entry for some people, but Myke has done an excellent job of easing new hardware developers through these problems by formulating this book as many short, simple experiments that incrementally build upon knowledge learned previously.

He starts out with so-called "experiments" that aren't experiments, but are really introductions to the development tools and the PIC MCU platform. Throughout the book, an emphasis is put on good design habits and early on he demonstrates techniques that will make the future development of large projects easier. A very thorough explanation of the MPLAB debugger is given, including several "experiments" which do nothing but explain how to see what's going on inside the MCU. At first I was critical that so much time was spent on the debugger, but after reading through this section, I'm glad that I now understand this powerful feature and expect it to drastically decrease my error rate.

Once you begin designing PIC applications, the C language is used, since its syntax is a little more like English than the commonly-used assembly language. This allows new developers to focus on the hardware platform.

Once several foundational experiments have been worked through, Myke begins teaching the use of Microchip's easy-to-learn (31 instruction) assembly language. All of the advanced experiments in this book use assembly language. This scared me at first, but Microchip has a brilliantly simple assembly language for the midlevel series of MCUs used in this book, and Myke's gentle introduction and easy-to-read writing style make this task palatable.

The hardware platform used is the well designed and well built Microchip PICKit 1 Flash Starter Kit, which is a USB device that acts as the programmer and test board. The book focuses on the 14 pin 16F684 MCU, but also uses the 8 pin 12F675 MCU on a few of the projects. While these microcontrollers are not the most feature packed in the PIC lineup, they do allow an excellent variety of projects and offer a number of useful features.

The 16F684 is the highest end product supported by the PICKit 1 programmer, but the package as a whole works exceptionally well, allowing budding developers to focus on the MCU and software aspects of design with no need to waste time figuring out how to get the individual components working together.

The PICKit 1 Flash Starter Kit is very affordable — costing only \$36 — and, as a bonus, the book contains a 20% discount coupon on the last page. Please note that your kit probably does not come with the 16F684 MCU, so order one of those in addition to your starter kit.

Some interesting experiments performed while working through this book include PC to MCU communication, the use of seven segment LED displays, and LCDs using the standard eight-bit protocol, four-bit protocol, and even a unique two-bit protocol. Additionally, you will create a digital "piano," interface with various sensors, and you will learn to control several types of motors including a DC motor, servo motor, and a stepper motor. The final group of projects convert a miniscule ZipZap remote controlled car into an autonomous line-following robot. I can't wait to do this experiment; I was even fortunate enough to purchase one of last year's ZipZap cars on clearance for \$12.50!

Microchip's development tools are designed to work on Windows, however, the C compiler used in this

book works on Linux or Windows and the source code for the USB programmer is included on the PICkit 1 CD-ROM. Microchip's official forums include a how-to including screenshots for configuring the MPLAB IDE under Wine. In essence, everything *should* work fine under Linux/Wine.

If I were going to gripe (and I really try not to), my biggest complaint is that a number of miscellaneous components are needed to perform the more advanced experiments. This is extremely frustrating if you have to wait four to five days to get the parts. It would be nice to have a consolidated parts list so that I could make one order to a discount parts company and get all of the parts.

One final note is that electronics fundamentals and theory are not explained in detail. Everything you need to know to build the example circuits is included in the book, but there is not a lot of explanation about the theory of operation outside of the MCU and code.

I highly suggest picking up this book if you would like to venture into the world of microcontrollers. It is easy to read, broken into small bite-sized experiments, and includes detailed illustrations and code. A basic understanding of electronics will help, but the author has brilliantly written the book in order to lower the barrier to entry. *123 PIC Microcontroller Experiments for the Evil Genius* will make an excellent addition to the bookshelf of any technical enthusiast who is unafraid of a soldering iron. SV

## Come One, Come All

If you are interested in adding *123 PIC Microcontroller Experiments for the Evil Genius* to your bookshelf, you can order it from the *SERVO* bookstore. Just call us at 800-783-4624 or order online from the *SERVO* website at **www.servomagazine.com**

# The Thereping

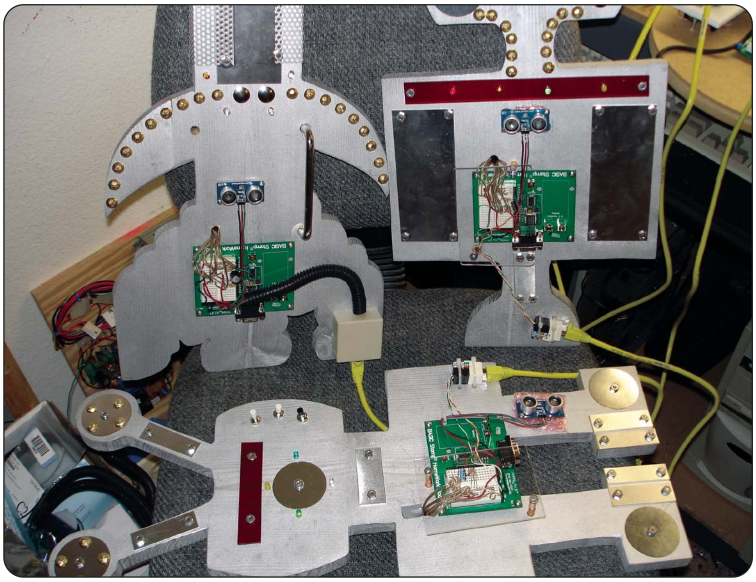
Vern Graner

The quest for a new and interesting musical instrument began when The Robot Group (an Austin, TX-based Robotics, Art, and Technology club) was selected to participate in the First Night Austin celebration with a proposal they entitled "The Robot Theremin Band." The idea was to use Theremins and robots together to create an experimental, musical spectacle. However, the Theremin is a difficult instrument to play *well* as it requires a high degree of accurate physical control (i.e., hand/body position), as well as a good knowledge of music theory. Thus, the Thereping was born.

The "Thereping" is an experimental musical instrument using a microcontroller and a sonar sensor to produce music. The device borrows operational concepts from the Theremin, the guitar, and even the bagpipes. The device is "worn" in the form of a guitar and played with two hands. A sonar sensor is used to determine the player's hand position over the sensor and then selects the note pitch (similar to a Theremin) while the left hand is used to press buttons that select the duration of the notes played. One button is singled out to allow the instrument to play a drone (similar to the bagpipes) so the player can rest but still contribute to the music around them.

Each Thereping is designed around a Basic Stamp II HomeWork board that is used to sample the sonar sensor and restrict the notes to valid choices in a blues scale to make sure the player cannot play a wrong note. Each Thereping is connected to a central "Thereclock" that provides power to the instrument and also sends sync pulses so all the instruments play together in tempo. The Thereclock also sends MIDI commands to a drum module to provide a beat to accompany the players. A construction article on the Thereping is available in the April issue of *Nuts & Volts Magazine* (**www.nutsvolts.com**). For details, construction information, or to interact with other Thereping players and builders, please visit the Thereping website at **www.thereping.com**


Send us a high-res picture of your robot with a few descriptive sentences and we'll make you famous. Well, mostly. menagerie@servomagazine.com



# THE ASSEMBLY LINE

by James Antonakos

## Uno Takes a Bow

Since August 2004, I have described my personal challenge to design, build, and troubleshoot my first robot. The 10 articles that resulted from my experience detailed the process from several points of view.

First came the planning. What was I going to try to build? My inspiration came from a 1950's *Scientific American* article that described a simple vacuum tube robot that responded to changes in light.

Next, I looked at the robot as a system. What were the functions required? How would they be accomplished with hardware and software? Having never designed a robot before, this was an interesting challenge.

Once I had an idea of what I needed in the design, I began purchasing and testing the parts: photocell, tilt switch, DC motor and relays, and the stepper motor. The entire system was constructed on a prototyping board for testing.

Using the BASIC Stamp II, I then wrote the necessary code to operate the electronics, proving the validity of the design.

Then, I built Uno's chassis and moved all the electronics and mechanical assemblies onto it. My first application of power resulted in lots of smoke from the stepper motor driver IC, which led to a troubleshooting investigation. Two incorrect wires (power and ground, unfortunately) were the culprit, and after fixing them, the smoke went away and Uno's simple control program ran the motors correctly.

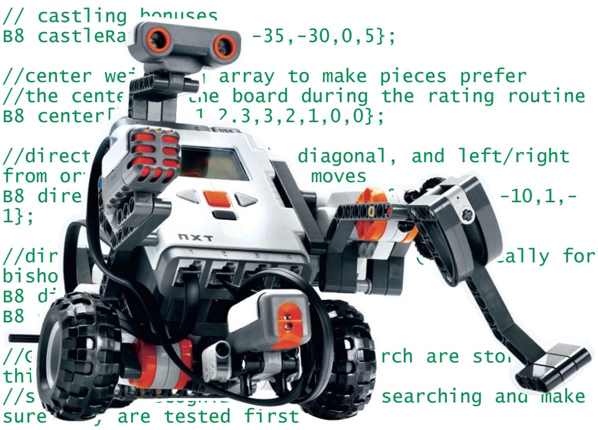
Now, a little more than 18 months after beginning the Uno project, I have a working robot ready to fully mimic the behavior of the 1950's robot that inspired it.

And this is where Uno's story takes a turn. Due to editorial decisions, "The Assembly Line" is ending its run with this installment. Those that have followed Uno's creation will be able to follow future work on the robot by visiting Uno's website located at **http://web.sunybroome.edu/~antonakos_j/uno**

This project has been an extremely rewarding learning experience for me, and I thank the editors at *SERVO Magazine* for making it possible. If I was able to demonstrate anything, it was that with a little effort and a dream, your ideas can become reality, with a good helping of satisfaction along the way. **SV**

### ABOUT THE AUTHOR

*James Antonakos is a Professor in the Departments of Electrical Engineering Technology and Computer Studies at Broome Community College, with over 27 years of experience designing digital and analog circuitry and developing software. He is also the author of numerous textbooks on microprocessors, programming, and microcomputer systems. You may reach him at antonakos_j@sunybroome.edu or visit his website at **www.sunybroome.edu/~antonakos_j***




# LESSONS FROM THE LABORATORY


## Look What's Coming ... NXT!

by James Isom

In January of this year, after years of speculation from LEGO robotics fans everywhere, LEGO announced plans for a new addition to its Mindstorms robotics line. Officially called LEGO Mindstorms NXT, the sets consist of all new hardware and software packing a host of new features and capabilities guaranteed to channel the inner nine-year-old in all of us.

Over the next couple of articles, we will explore this system in anticipation of its release this August.

### The NXT

Table 1 shows a run-down of a few of the RCX features as compared with new NXT features.

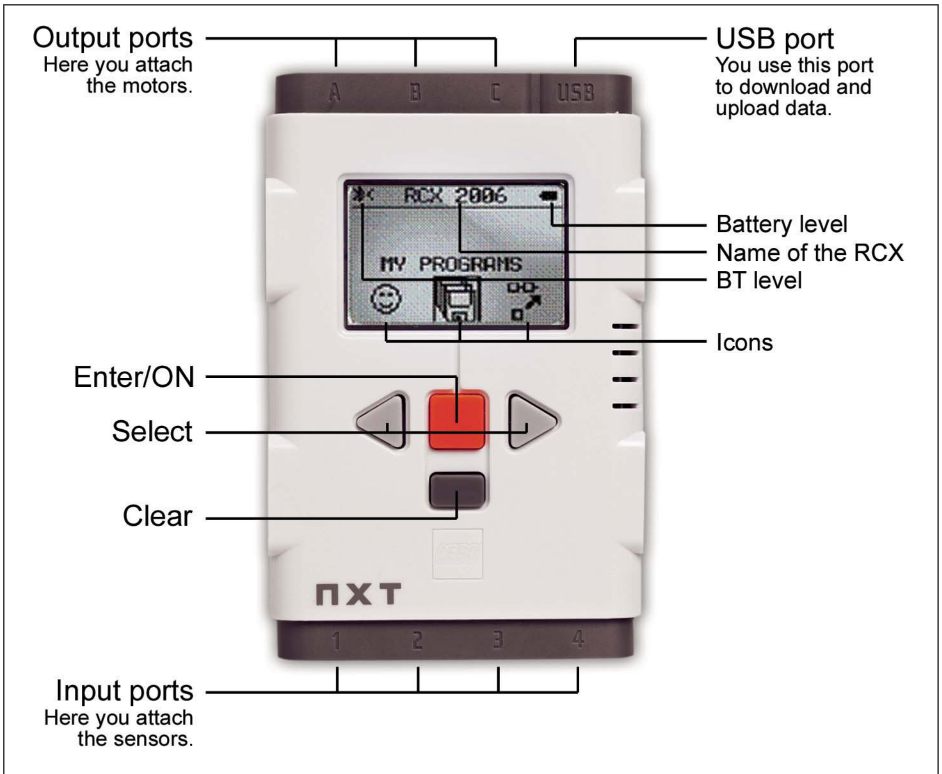
Let's start by looking at the new NXT programmable brick. Approximately the same size as its yellow cousin — the RCX — the similarities between the new NXT and the RCX pretty much end there. The first thing you'll notice is the lack of the 2 x 2 sensor pads that formerly served as the RCX's input and output ports. The sensors now attach to the top and bottom of the NXT via connectors similar to phone jacks.

The bottom has room for four sensors and the top sports three power ports, leaving room for the new USB 2.0 port. The USB port replaces the infrared tower as one of two ways to transfer your program from the computer to your robot. Communication with the host computer or additional NXTs can also take place using Bluetooth wireless technology.

The face of the NXT is dominated by the new 60 x 100 pixel dot matrix display. Navigating between programs

**TABLE 1**

| Feature | RCX | NXT |
|---|---|---|
| Processor | Eight bit Hitachi Microprocessor | 32 bit Arm7 Microprocessor |
| Communication | Infrared | Bluetooth/USB 2.0 |
| Inputs | 3 | 4 |
| Outputs | 3 | 3 |
| Display | Five digit LCD + the running man | 60 x 100 pixel programmable dot matrix display |
| Programming Space | Five program slots | As many as will fit in 245K of Flash |
| Power | Six AA batteries | • Six AA batteries<br>• Rechargeable Lithium battery with AC port (educational) |




The NXT and its parts.

is as simple as scrolling through songs on your favorite MP3 player — the days of having just five program slots to play with are gone. Not only does it allow you to easily navigate between programs, but the NXT also allows you to make simple programs right on screen without the need to be at a computer. You won't be doing any major programming this way, but it's great for a quick demonstration or proof of concept.

The new display can also show simple graphics that can be controlled directly from your program. For example, your robot could roam around the room while displaying a pair of eyeballs looking to and fro. The buttons on the NXT are also programmable. For example, your program can now wait for the left select button to be pressed. The battery pack replaces the bottom of the NXT brick

Like the RCX, the NXT will be powered by six AA batteries. However, if you choose to purchase the educational version of the kit, it will include the rechargeable battery pack option that fits in the same space as the six AA batteries. AAs or rechargeable battery? It's up to you, but the rechargeable Lithium battery has a minimum capacity of 670% over the AA option and charges in four hours.

However, because it is a Lithium battery, it will work with a partial charge and will never be in danger of overcharging (and psst! — the rechargeable option is better for the environment). The battery pack also includes an AC adapter jack allowing you to run from direct power for those long term experiments or crazy ball contraptions.

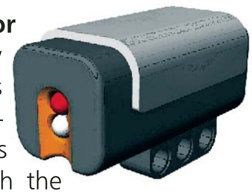
## The Sensors

The current LEGO robotics system supports four sensors: touch, light, rotation, and temperature. The new NXT system supports most of the same, along with a couple of new ones.

**The Touch Sensor**

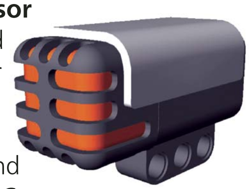
The touch sensor is pretty much the same as the previous touch sensor. The big difference is the addition of an axle hole on the touch pad.

**The Light Sensor**



The new light sensor is much more sensitive than its older cousin with the ability to pick up finer gradations of light than ever before. A problem with the previous light sensor was that if you were trying to read ambient light values, it would sometimes be influenced by its own light source. This has been solved in the new version with the ability to turn the light source on and off from within your program.

**The Sound Sensor**



The sound sensor is new for the NXT and will allow your robot to sense tones and sound patterns. Can anyone say "Clap on! Clap off!?" Two claps for right, one for left? There are lots of fun possibilities here.

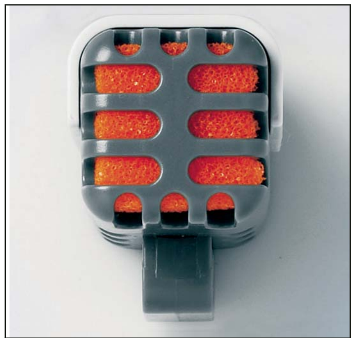
**The Ultrasonic Sensor**

Ultrasonic sensors have long been a staple of the hobby robotics realm, and now the NXT has one, too. You'll be able to sense objects at a specific distance or react to movement in the room. This will make navigating a room without ever touching an object finely a possibility.

Now, many of you might be saying, "Hey where's the temperature and rotation sensor?" LEGO is not releasing a


Touch sensor.


Light sensor.


Sound sensor.


Ultrasonic sensor.

new temperature sensor at this time, although I am sure a third party solution won't be long to market. If you absolutely need a temperature sensor for your application, you can use the converter cables that will be available in the educational sets. These cables can be used with any of the previous motors and sensors to interface with the NXT brick.

Even if have a few of the old rotation sensors laying around, you might not always need them with the NXT. The new NXT motors have rotation sensors built into them and are actually servos. A servo is a motor that automatically adjusts itself to move a certain amount. This allows very precise movements, giving you much more accuracy than the previous system ever allowed and, now that each motor has a rotation sensor built in, not only will it not take up a valuable input port, but with sensors in each motor, it will be far easier to make your robot drive in a straight line.

## The Software

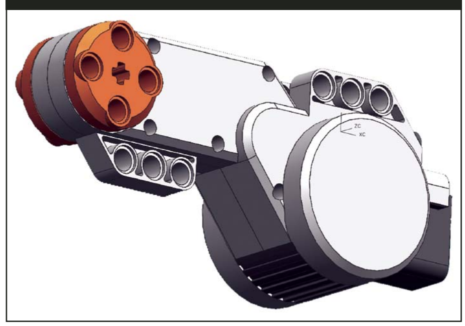
The new NXT software looks great — easy to use, easy to learn, with a robust set of features. It will run on PCs running Mac OSX or Windows. I'm not going to go too deep into it at this point because I'll be spending a whole article on this in the coming months.

For those of you who can't live without your Robolab, you will be happy to know that there is a 3.0 version planned. Robolab 3.0 will be compatible with both the RCX and the NXT. Rumor has it that it should be available about the same time the NXT hits the street, but don't tell anyone I told you.

For more information on the new system, check out the Mindstorms NXT blog at **http://legoeducation.typepad.com/**

I'll be back in June with more on the new Mindstorms NXT robotics system. Until then, have fun! **SV**

The motor.



An RCX to NXT conversion cable.

The battery pack replaces the bottom of the NXT brick.

NXT motors and sensors.

Robotics Resources

*Tune in each month for a heads-up on where to get all of your "robotics resources" for the best prices!*

By Gordon McComb

# Radio Frequency Identification

RFID stands for radio frequency identification — a kind of sensor that is similar in purpose to barcodes, but is meant to operate over longer distances, even through other objects. (Implantable biochips, like the kind used for pets — and now people — are miniature RFID units.) RFID uses small devices that radiate a digital signature when exposed to a radio frequency signal. RFID is found in products ranging from toys to employee access cards, to gasoline pump "key fobs" and trucking, to farm animal inventories, automobile manufacturing, and more.

In this installment of Robotics Resources, we'll look at the exciting field of RFID, and it how it can be applied to amateur robotics.

## Under the Hood — How RFID Works

A transmitter/receiver, called the *interrogator* (also the *reader*, *transponder*, or *host*), radiates a low- or medium-frequency carrier RF signal. If it is within range, a passive (unpowered) or active (powered) detector, called a *tag* or *transponder*, re-radiates (or backscatters) the carrier frequency, along with a digital signature that uniquely identifies the device. RFID systems in use today operate on several common RF bands, including a low-speed 100-150 kHz band typified by the TIRIS line from Texas Instruments, and a higher 13.5 MHz band. Other products operate in the 900 MHz UHF band, and the 2.45 and 5.8 GHz microwave bands.

The tag is composed of an antenna coil along with an integrated circuit. The radio signal provides power when used with passive tags, using well-known RF field induction principles. Inside the integrated circuit are decoding electronics and a small memory. A variety of data transmission schemes are used, including non-return to zero, frequency shift keying, and phase shift keying.

Manufacturers of the RFID devices tend to favor one system over another, depending on the intended application. Some data modulation schemes are better at long distances, for example.

## Why Use RFID?

How can they be used in robotics? Here are a few ideas:

- Place RFID tags on multiple robots that are meant to work together. The robots would be able to identify one another as they come in proximity.

- Place RFID tags on people. This would allow a robot, or group of robots, to identify each person.

- Place RFID tags along baseboards of rooms or door jambs. As a robot passes by a tag, it can determine where in a room, even which room, it is currently located in.

It is the last idea that holds much promise for amateur robots. Navigating a robot within a structure or yard is made even more complicated if there are several rooms or areas that the robot must be kept aware of. RFID tags allow the robot to determine which room or area it is currently occupying, without resorting to more complicated mapping or image recognition.

When used for room identification, RFID tags operate as a kind of lighthouse, orienting the robot as it travels. Besides RFID, there are several other ways to provide active navigation signals to a robot; one is infrared beacons placed strategically in a room or area. Surprisingly, the advantage of RFID over infrared beacons is that the coverage of the RF signal is naturally limited. This provides a convenient way to differentiate the areas of a house's robotic work space.

## Memory and Cost

RFID tags have differing amounts of memory, from just a few bytes to several thousand bytes. Most have 32 to 128 bytes — enough to store a serial number, date, and other limited data. For robotics, this is more than enough to serve as room-by-room, or locale-by-locale beacons.

While RFID systems are not complex, with few exceptions, cost is not yet in the super-affordable region. Demonstration and developer's kits are available from some manufacturers in the \$200-\$300 range, and this

includes the reader and an assortment of tags.

Hand-held interrogators cost between $150 and $500; the tags cost under $1 each, but most manufacturers want you to buy them in quantities of 250+, making RFID an expensive proposition. However, once implemented, RFID is a carefree and long-term solution to helping your robot know where it is.

Among the least expensive RFID solutions available today is the Grand Idea Studio reader and tags, sold by Parallax. A reader and assortment of tags costs under $50, and the reader can be interfaced to the BASIC Stamp and other low-cost microcontrollers via a serial line. Each transponder tag is precoded with a unique numeric value. See the BASIC Stamp column in the April '05 issue of *Nuts & Volts* for more information on this product, including code examples (**www.nutsvolts.com**).

Also available is a USB-based RFID experimenter kit from Phidgets USA. The company sells a wide assortment of tag styles, from small discs to credit card size. A full kit — with reader and a variety of tags — sells for $80.

## Using Complementary Sensors With RFID

Consider the human body: it has many kinds of "touch receptors" embedded within its skin. Some receptors are sensitive to physical pressure, while others are sensitive to heat. Similarly, you can produce robots that enjoy many senses, in addition to the positional information that RFID provides. Consider adding one or more of these sensors to your robots:

• *Heat sensors* can detect changes in the heat of objects within grasp. Heat sensors are available in many forms, including thermisters (resistors that change their value depending on temperature) and solid-state diodes that are specifically made to be ultra-sensitive to changes in temperature.

• *Air pressure sensors* can be used to detect physical contact. The sensor is connected to a flexible tube or bladder (like a balloon); pressure on the tube or bladder causes air to push into or out of the sensor, thereby triggering it. To be useful, the sensor should be sensitive down to about one pound per square inch, or less.

• *Resistive bend sensors*, originally designed for use with virtual reality gloves, vary their resistance, depending on the degree of bending. Mount the sensor in a loop, and you can detect the change in resistance as the loop is deformed by the pressure of contact.

• *Strain gauges* measure the stress on parts of the robot. You can mount strain gauges on the front and back bumpers of a robot, for example, and they will detect when the 'bot has collided with an object. Strain gauges that detect a DC bias (in other words, a constant pressure) can be used to determine if the robot remains pressed against an object. Strain gauges that cannot detect a DC bias (piezoelectric cells are included in this group) can only determine change between contact and non-contact.

### ROBOTS THAT FOLLOW WALLS

By their design, the typical RFID tag must be fairly close to its reader in order to receive and reflect a signal. This avoids problems of interference, but it also means that in order for your robot to reliably use RFID, it must position itself as close to the transponders as possible. Otherwise, the RFID tag may never come close enough to the reader to produce a readable signal.

One approach to this problem is to use readers and tags with a larger signal area. You must be extra careful that you position the tags (and/or readers) far enough away from one another so that there is little chance their signals will interfere with one another.

Another way to tackle the problem, while avoiding many of the issues of signal interference, is to intentionally drive the 'bot near where the tag is located. This can be accomplished by using a wall-following behavior

Mice exhibit a similar behavior: when indoors, they tend to run along the sides of walls, rather than out in the center of the room. This is for their protection. Mice are prey, and they can be more easily seen in the open.

A common robot experiment is to construct a wall-following "mouse" — a mechanical rodent that favors walls, and even actively seeks them out before exploring the rest of the room. Most all wall-following robots use a touch or proximity sensor to detect the wall. There are several common ways of accomplishing this:

• *Whisker contact.* The robot uses a mechanical switch, or a stiff wire that is connected to a switch, to sense contact with the wall. This is by far the most simple method, but is prone to mechanical damage after a period of use.

• *Non-contact, active sensor.* The robot uses active proximity sensors, such as infrared or ultrasonic, to determine distance from the wall. No physical contact with the wall is needed. In a typical non-contact system, two sensors are used to judge when the robot is parallel to the wall.

• *Non-contact, passive sensor.* The robot uses passive sensors, such as linear Hall-effect switches, to judge distance from a specially-prepared wall. In the case of Hall-effect switches, the baseboard or wall might be outfitted with an electrical wire, through which a low-voltage alternating current is fed. When in proximity, the sensors will pick up the induced magnetic field provided by the alternating current. Or, if the baseboard is metal, the Hall-effect sensor (when rigged with a small magnet on its opposite side) could detect proximity to a wall.

• *"Soft-contact."* The robot uses mechanical means to detect contact with the wall, but the contact is "softened" with the use of pliable materials. For example, a lightweight foam wheel can be used as a "wall roller." The benefit of soft contact is that mechanical failure is reduced or eliminated, because the contact with the wall is through an elastic or pliable medium.

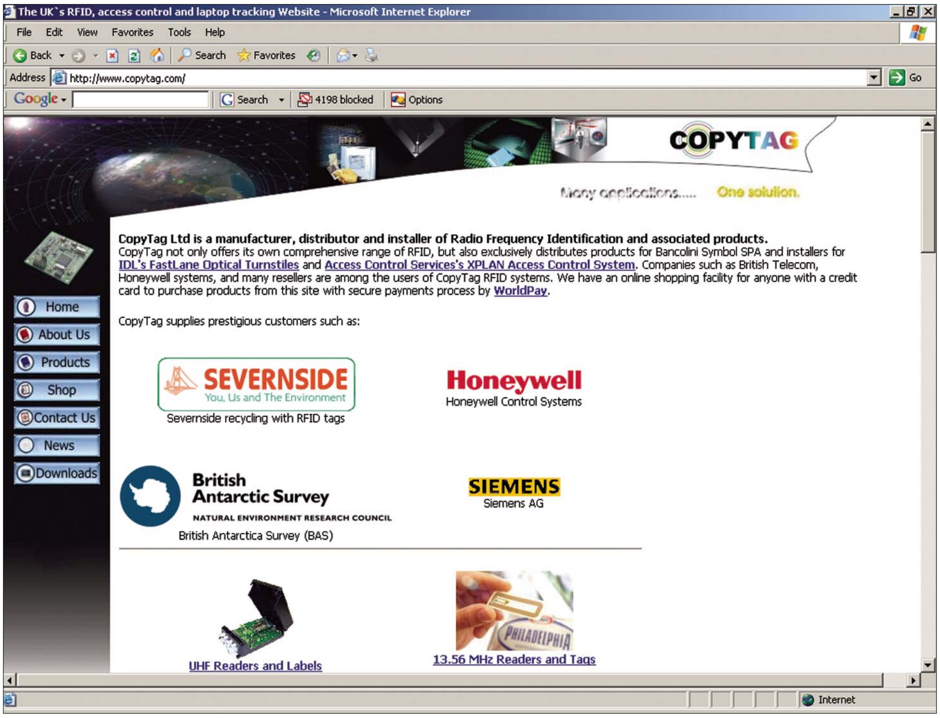



**FIGURE 1.** CopyTag is one of several online resources for commercial and industrial RFID solutions.

• *Microphones and other sound transducers* make effective touch sensors. Microphones — either standard or ultrasonic — can be used to detect sounds that occur when objects touch ("microphonic conduction," for the lack of a better term). Mount the microphone element on the robot. Place a small piece of felt directly under the element, and cement it in place using a household glue that sets up hard. Run the leads of the microphone to a sound trigger circuit. As things move past the sensor, it will pick up the sound.

## ELECTRONIC COMPASSES AND RFID

Radio frequency ID tags provide a means to determine the general location of a robot within a confined space, but they don't indicate orientation. Knowing orientation is a prerequisite to effective navigation. Just like sea and land explorers of old, robots can use magnetic compasses for navigation, and a number of electronic and electromechanical compasses are available for use in hobby robots. The most basic compasses are accurate to about 45 degrees, and provide heading information (N, S, E, W, SE, SW, NE, NW) by measuring the Earth's magnetic field.

One of the least expensive electronic compasses is the Dinsmore 1490, from Dinsmore Instrument Co. The 1490 detects the Earth's magnetic field by using miniature Hall-effect sensors and a rotating compass needle (similar to ordinary compasses). The sensor is said to be internally designed to respond to directional changes similar to a liquid filled compass, turning to the indicated direction from a 90 degree displacement in approximately 2.5 seconds.

Dinsmore Instrument Co.
**www.dinsmoresensors.com**

Another option is the Compass Module from Robot-Electronics, which is resold by a number of online retailers, such as Acroname. With a stated accuracy of ±3-4°, this compass uses a variety of interface methods, including a digital pulse train, or an I$^2$C serial network.

Robot-Electronics.com
**www.robot-electronics.com**

## Sources

Applications in robotics are both obvious and numerous: you can use RFID for robot-to-robot identification, robot-to-human identification, navigation, beacon systems, and much more. A benefit of RFID is that the sensitivity of the reader electronics can be varied, so that you can directly control maximum working distances. In this way, a room could be full of RFID elements, yet your robot will only "see" the one closest to it.

As yet, there are few RFID systems within affordable reach of most amateur robot builders; still, it's an interesting technology, and it's only a matter of time (perhaps just months) before affordable entry-level solutions become readily available. If nothing else, you can use the follow resources to learn more about this technology.

### *CopyTag Limited*
**www.copytag.com**

Makers of radio frequency identification (RFID) readers and tags (transponders).

### *Microchip Technology*
**www.microchip.com**

In addition to its broad line of semiconductors — including the venerable PICmicro microcontrollers — Microchop is also involved with RFID, selling readers and tags, as well as developer's kits.

### *OMRON Corp*
**www.omron.com**

Omron is a multi-disciplinary company, manufacturing a wide array of sensors and semiconductors. Of note are their RFID tags and readers, and machine vision products.

### *Parallax*
**www.parallax.com**

Offers a low-cost ($40) reader module, as well as several variations of tags, which they sell in any quantity. Naturally, the company provides documentation on using the reader module with their venerable BASIC Stamp modules.

### *Phidgets USA*
**www.phidgetsusa.com**

USB-based RFID readers and tags. Designed for the hobbyist market.

### *RACO Industries/ID Warehouse*
**www.idwarehouse.com**

Resellers of various barcode and RFID tagging systems.

### *RFID Components Ltd.*
**www.rfid.co.uk**

RFID Components Limited is a specialist distributor of products for automatic recognition and identification applications. Their products include the Texas Instruments TI*RFID TIRIS line.

### *RFID, Inc.*
**www.rfidinc.com**

Makers and sellers of RFID receivers and transponder tags. Offers relatively inexpensive starter kits with sampler tags and receiver.

### *RFID.org*
**www.rfid.org**

Everything about radio frequency identification. Sponsored by AIM, the global trade association for the Automatic Identification and Data Capture industry.

### *RFID Journal*
**www.rfidjournal.com**

Online industry trade publication on all things RFID. A print magazine is also available. Includes a number of useful (and free) white papers on RFID technology.

### *Unified Barcode*
**www.unifiedbarcode.com**

Online reseller of barcode and RFID solutions.

## ABOUT THE AUTHOR

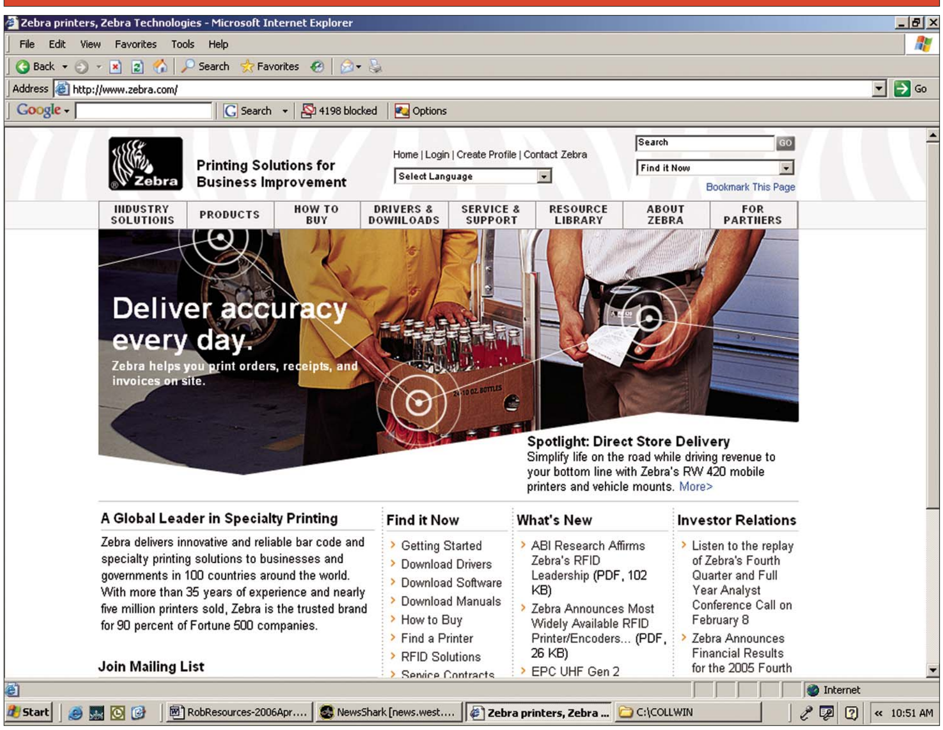
*Gordon McComb is the author of the best-selling Robot Builder's Bonanza, Robot Builder's Sourcebook, and Constructing Robot Bases — all from Tab/McGraw-Hill. In addition to writing books, he operates a small manufacturing company dedicated to low-cost amateur robotics, **www.budgetrobotics.com** He can be reached at robots@robotoid.com*


**FIGURE 2.** Among its numerous interesting product lines, Phidgets USA offers low-cost RFID components and kits.

### *Zebra Technologies Corporation*
**www.zebra.com**

Zebra Technologies Corporation is a manufacturer of barcodes, readers (wands, CCD, and laser scanners), RFID readers and tags, and barcode label printers. **SV**

**FIGURE 3.** Zebra Technologies is well known in the barcode field. They provide a number of ready-to-go RFID products, as well.

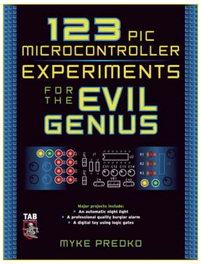


Mind Candy For Today's Roboticist

# The SERVO Store

## 123 PIC Microcontroller Experiments for the Evil Genius

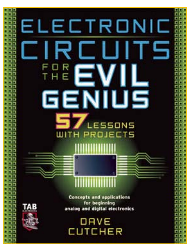
by Myke Predko




Few books take advantage of all the work done by Microchip. *123 PIC Microcontroller Experiments for the Evil Genius* uses the best parts, and does not become dependent on one tool type or version, to accommodate the widest audience possible. Building on the success of *123 Robotics Experiments for the Evil Genius*, as well as the unbelievable sales history of *Programming and Customizing the PIC Microcontroller*, this book will combine the format of the evil genius title with the following of the microcontroller audience for a sure-fire hit. **$24.95**

## Electronic Circuits for the Evil Genius

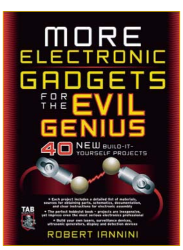
by Dave Cutcher




Cutcher's 57 lessons build on each other and add up to projects that are fun and practical. The reader gains valuable experience in circuit construction and design and in learning to test, modify, and observe results. Bonus website **www.books.mcgraw-hill com/authors/cutcher** provides animations, answers to worksheet problems, links to other resources, WAV files to be used as frequency generators, and freeware to apply your PC as an oscilloscope. **$24.95**

## MORE Electronic Gadgets for the Evil Genius

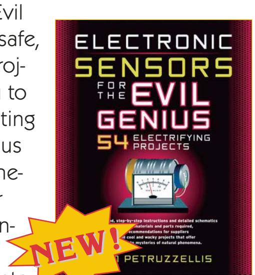
by Robert E. Iannini




This much anticipated follow-up to the wildly popular cult classic *Electronic Gadgets for the Evil Genius* gives basement experimenters 40 all-new projects to tinker with. Following the tried-and-true Evil Genius Series format, each project includes a detailed list of materials, sources for parts, schematics, documentation, and lots of clear, well-illustrated instructions for easy assembly. Readers will also get a quick briefing on mathematical theory and a simple explanation of operation. **$24.95**

## Electronic Sensors for the Evil Genius — 54 Electrifying Projects

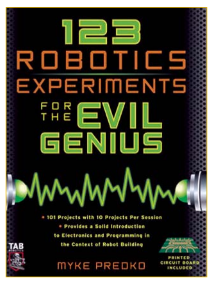
by Thomas Petruzzellis




Nature meets the Evil Genius via 54 fun, safe, and inexpensive projects that allow you to explore the fascinating and often mysterious world of natural phenomena using your own home-built sensors. Each project includes a list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions. Projects include rain detector, air pressure sensor, cloud chamber, lightning detector, electronic gas sniffer, seismograph, radiation detector, and much more! **$24.95**

## 123 Robotics Experiments for the Evil Genius

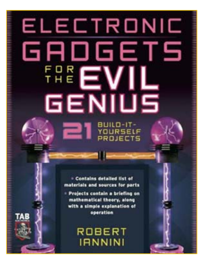
by Myke Predko




If you enjoy tinkering in your workshop and have a fascination for robotics, you'll have hours of fun working through the 123 experiments found in this innovative project book. More than just an enjoyable way to spend time, these exciting experiments also provide a solid grounding in robotics, electronics, and programming. Each experiment builds on the skills acquired in those before it so you develop a hands-on, nuts-and-bolts understanding of robotics — from the ground up. **$25.00**

## Electronic Gadgets for the Evil Genius

by Robert Iannini

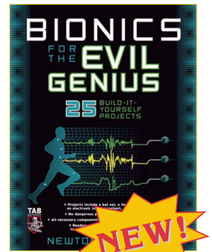



The do-it-yourself hobbyist market — particularly in the area of electronics — is hotter than ever. This book gives the "evil genius" loads of projects to delve into, from an ultrasonic microphone to a body heat detector, all the way to a *Star Wars* Light Saber. This book makes creating these devices fun, inexpensive, and easy. **$24.95**

**We accept VISA, MC, AMEX, and DISCOVER**
**Prices do not include shipping and may be subject to change.**

## Bionics for the Evil Genius

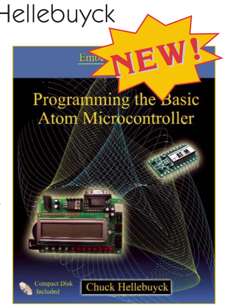
by Newton C. Braga




Step into the future (or the past, if you have a touch of Dr. Frankenstein in your soul) with these 25 incredibly cool bionic experiments! Demonstrating how life forms can be enhanced, combined, manipulated, and measured with electronic and mechanical components, these inexpensive projects from internationally renowned electronics guru Newton Braga provide hours of fun and learning. Totally safe, *Bionics for the Evil Genius* guides you step-by-step through 25 complete, intriguing, and low-cost projects developed especially for this book — including an electric fish, a bat ear, a lie detector, an electronic nerve stimulator, a panic generator, and 20 other exciting bioelectric/mechanical projects!**$24.95**

## Programming the Basic Atom Microcontroller

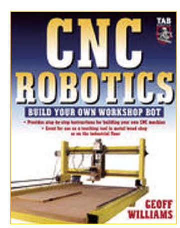
by Chuck Hellebuyck




Through his unique way of making the complicated understandable, Chuck takes the reader through the inner workings of the Basic Atom by explaining the Microchip PIC Microcontroller and its roll in the Atom module. From there, Chuck explains the various PIC based Basic Atom modules and how to use the Basic Atom compiler. Chuck then delivers 13 projects the reader can build and learn from. The reader can then use this knowledge to develop their own Basic Atom projects. **$39.95**

## CNC Robotics

by Geoff Williams

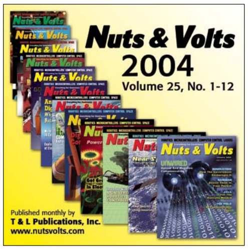



Now, for the first time, you can get complete directions for building a CNC workshop bot for a total cost of around $1,500.00. *CNC Robotics* gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot and can be customized to suit your purposes exactly, because you designed it. **$34.95**

**To order call 1-800-783-4624 or go to our website at www.servomagazine.com**

**Mind Candy For Today's Roboticist**

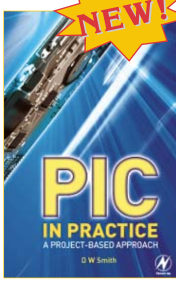
## *Nuts & Volts* CD-Rom




Here's some good news for *Nuts & Volts* readers! Starting with the January 2004 issue of *Nuts & Volts*, all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 25, issues 1-12, for a total of 12 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

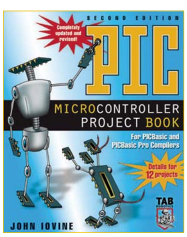
## PIC in Practice
### A Project-based Approach Second Edition
by David W. Smith

NEW!




*PIC in Practice* is a graded course based around the practical use of the PIC microcontroller through project work. Principles are introduced gradually, through hands-on experience, enabling students to develop their understanding at their own pace. The book can be used at a variety of levels and the carefully graded projects make it ideal for colleges, schools, and universities. Newcomers to the PIC will find it a painless introduction, while electronics hobbyists will enjoy the practical nature of this first course in microcontrollers. **$29.95**

## PIC Microcontroller Project Book
by John Iovine



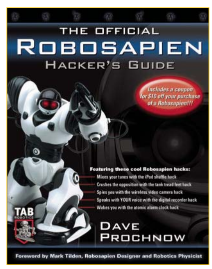

The PIC microcontroller is enormously popular both in the US and abroad. The first edition of this book was a tremendous success because of that. However, many users of the PIC are now comfortable paying the $250.00 price for the Professional version of the PIC Basic. This new edition is fully updated and revised to include detailed directions on using both versions of the microcontroller, with no-nonsense recommendations on which one serves better in different situations. **$29.95**

**Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.**

## *SERVO* CD-Rom

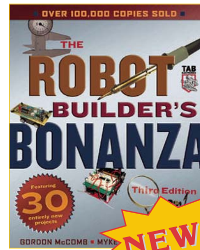

Are you ready for some good news? Starting with the first *SERVO Magazine* issue — November 2003 — all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 1, issues 11-12 and Volume 2, issues 1-12, for a total of 14 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

## The Official Robosapien Hacker's Guide
by Dave Prochnow




The Robosapien robot was one of the most popular hobbyist gifts of the 2004 holiday season, selling approximately 1.5 million units at major retail outlets. The brief manual accompanying the robot covered only basic movements and maneuvers — the robot's real power and potential remain undiscovered by most owners — until now! This is the official Robosapien guide — endorsed by WowWee (the manufacturer) and Mark Tilden (the designer). This timely book covers possible design additions, programming possibilities, and "hacks" not found any place else. **$24.95**

## Robot Builder's Bonanza Third Edition
by Gordon McComb / Myke Predko




NEW!

Everybody's favorite amateur robotics book is bolder and better than ever — and now features the field's "grand master" Myke Predko as the new author! Author duo McComb and Predko bring their expertise to this fully-illustrated robotics "bible" to enhance the already incomparable content on how to build — and have a universe of fun — with robots. Projects vary in complexity so everyone from novices to advanced hobbyists will find something of interest. Among the many new editions, this book features 30 completely new projects! **$27.95**

## Linux Robotics
by D. Jay Newman


Now Shipping!

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need, plus complete training in doing original programming. **$34.95**

$20 each
15% off if you buy two or more.


From HomoSapien to RoboSapien


Before R2D2 there was R1D1

## *SERVO Magazine* T-Shirts

Get your very own limited-edition *SERVO Magazine* T-shirt. Shirts come in sizes S, M, L, and XL, and are available in either black or white. All shirts are 100% pre shrunk cotton.

# ROBO-LINKS

ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies Since 1967
Save up to 60% on electronic components and microcontrollers
www.futurlec.com
RobotShop
Robotics at your service!™
www.robotshop.ca
Robot Building Blocks!
Sensors, Servos, Smarts, Software, Kits
www.garage-technologies.com
Affordable DC motors with encoders
Amplifiers and Controllers for motors
Encoder/Motor related parts and goodies
Look/See Encodergeek.com
Give your Robot the power it deserves...
LORAX WORKS
proSeed microcontroller
www.loraxworks.com
For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links to hotlink to these great companies.
MeerKat Systems, Inc.
Home Page http://www.meerkatsystems.net
Interfacing Kits for USB, RJ11, RJ12, and Firewire
Handheld Pendants for Prototyping
LCD Mounting Plates
Visit our eBay Store at http://stores.ebay.com/MeerKat-Systems
NOW IS THE TIME
To order your subscription to SERVO Magazine! Call us TODAY or visit our website to place your order.
877.525.2539 www.servomagazine.com

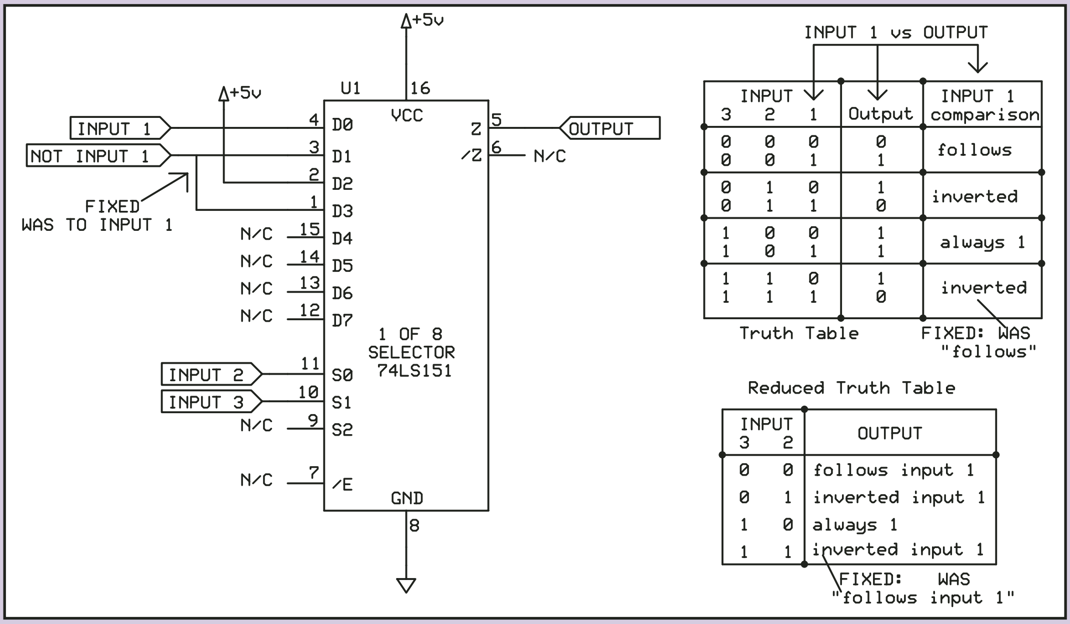

BUILDING ROBOTS IS FUN FOR EVERYBODY... EVERYWHERE!
RobotShop
WWW.ROBOTSHOP.CA

# BIO-FEEDBACK

*Continued from Page 7*

*is provided here. I hope that this clarifies matters and apologize for any confusion that was caused.*

**Gerard Fonte**




*Dear SERVO:*

Table 1 in the Mr. Roboto column in the February 06 issue gives the unit cell voltage for a Lithium battery as 3.0 volts. Alas, readers should know that there are different Lithium battery chemistries. While LiMnO2 is 3.0 volts, LiFeS2 is 1.5 volts. The latter is sold in AA sizes to replace Alkaline batteries. There may be other chemistries also; it would be good to get this all straightened out. I enjoy your column every month.

**John Piccirillo**

*Writer response:*

*Thanks for the information on the Lithium-Iron (LiFeS2) batteries that Energizer is now selling. I have been wondering how they get a 1.5 volt output from what is normally considered a three-volt technology. I contacted Energizer after I heard about their new batteries, but they wouldn't answer the question on how they get 1.5 volts. I guess we are going to start to see a lot more of the LiFeS2 batteries because of their high energy densities. For other people reading this, in addition to the 3.0 volts LiMnO2 technology, the rechargeable Lithium-Ion and Lithium-Polymer batteries output 3.6 volts. Thus, just mentioning a generic term "Lithium Battery" could be confusing to people since there are so many different voltage outputs from the family of Lithium batteries.*

**Pete Miles**
**Ask Mr. Roboto**

*Dear SERVO:*

I just finished reading David Geer's article about Jasper — the 3D Movie Bot in the March 06 issue.

While kudos to Mike Wilder for making his 3D time-lapse film, it seems to me that what he did with Jasper was create a very complex solution when a much simpler approach would have done an even better job. Instead of moving the camera back and forth with such precision for every stereo pair, he could have just left the camera fixed and used the turntable to rotate the plant a small amount to create the necessary parallax offset for stereoscopic imagery. This, in fact, would have produced better stereoscopic images since your eyes converge on objects, they do not stare ahead in parallel.

To further simplify, since his movies are of rotating plants, he could take advantage of the Pulfrich effect (Google it) and not even shoot stereo pairs. Pulfrich effect movies produce a very realistic 3D effect and do not require processing of stereo pairs or colored glasses to view them, just a dark filter over one eye. They also have the advantage of looking normal (no color fringes) when seen by someone without any special viewing apparatus.

**George Mitsuoka**

*Dear SERVO:*

I read the "Then and Now" column in the February 06 issue with interest, as it dealt with one of my focus areas — surgical robots. I'm very happy to see this area covered in *SERVO Magazine*, and hope to see additional articles on currently-marketed industrial robots that started out as the prototypes with which your readership is most familiar.

However, I was confused by Tom Carroll's presentation of Computer Motion, Inc. He made it seem like an existing, viable business. In fact, it was absorbed into Intuitive Surgical, Inc., almost three years ago, with a massive layoff. For a review of the situation that led to the disappearance of Computer Motion from the business scene, see **http://biomed.brown.edu/Courses/BI108/BI108_2005_Groups/04/davinci.html** (for example).

Readers interested in the AESOP, SOCRATES, and ZEUS systems that Computer Motion developed should consult Intuitive Surgical's website at **www.intuitivesurgical.com**

**Jim Curme**


**"I'll forever love the Aibo, but I see it for what it was — a sweet, old-fashioned king who must be replaced by a young, forward-thinking prince."** *— Mind/Iron, page 6*

**" ... Sony has decided to put Aibo — the robot dog — to sleep."** *— Robytes, page 9*


# AIBO — Sony Unleashes Feral Fido

***by Dave Prochnow***

The news couldn't have been more chilling or more foreboding. Buried on page 11 of the Sony Corporation Third Quarter FY2005 Consolidated Results presentation released on January 26, 2006 was this little paragraph (Q3 FY2005 Sony Group Earnings Announcement; **www.sony.net/SonyInfo/IR/financial/fr/index.htm**):

"Restoring Profitability in Specific Business Categories —

## Entertainment Robot

**New product development for AIBO has already ceased, and production is targeted to stop by the end of fiscal year 2005. However, after-sales support will continue. There will also be no new development for QRIO. R&D in the AI areawhich was developed for AIBO and QRIO businesses will continue and will be deployed in a broad range of consumer electronics products."**

Yes, in spite of showing a 4.7% sales increase for its third quarter, Sony closed its Entertainment Robot category. And so a chapter in robot history came to an unglamorous end. No fanfare, no special offers, and — get this — no photographic images of either AIBO or QRIO in the Sony Corporation historical imagery archive.

There you have it — the lives and times of AIBO and QRIO had been totally expunged from the annals of Sony Corporation's contributions to robotics.

Life wasn't always this acrimonious between Sony Corporation and its Entertainment Robot category.

Sony downright gushed over AIBO's 5th Anniversary back on May 6, 2004. Declaring this robot to have demonstrated "five years of innovation and intelligence." Oh, yeah, and you're fired, too. Don't let the door snag your tail on the way out.

Later in the same breath, Sony reprinted a quote attributed to Martin Newham, Founder and Leader of the AIBO Owner's Club in the UK. "After five years, it's good to see the concept is as fresh, fascinating, and inspiring as the day when it was first launched," Newham said.

This public love affair featured an odd juxtaposition with a "back channel" copyright infringement threat against the legendary Aibopet — a hacker with an enormous following among the roughly 150,000 AIBO owners of the $2,000 robot dog.

The problem stemmed from Aibopet's usage of some Sony copyrighted code that was embedded inside his own hacked software. In a 2002 statement, Sony spokesman, Mack Araki stated that, "we needed to have priority in protecting our copyright and we can't support the development of illegal software."

So the gauntlet was laid firmly at the feet of the formerly adoring public. They responded exactly the way any consumer would respond — they shunned Sony.

Now fast-forward to February 2006, when the Associated Press (AP) reported that many AIBO owners were disillusioned by Sony's announcement regarding the termination of the Entertainment Robot category. AIBO owners like Takeshi Ohashi — a Kyushu Institute of Technology professor and RoboCup organizer — thought that AIBO was a shining gem in Sony's crown demonstrating a high degree of "technological finesse."

Obviously living with denial, Ohashi plans to petition Sony to resus-

citate AIBO. Good luck.

## Out With the Old Boss

All you have to do is take one look at Sir Howard Stringer — Chairman and CEO of Sony Corporation — as he was delivering his keynote address at the 2006 International Consumer Electronics Show (CES) in Las Vegas, NV, and you know that this type of reincarnation is *not* going to happen.

In a new business category that sounded like something that was thought up the night before, called *e-Entertainment*, Sir Stringer stressed that Sony was "focused on the increasingly personalized nature of entertainment and the importance of recognizing and accommodating the needs of the individual while providing choice and convenience in the ways that consumers use Sony products."

It's tough to imagine a "personalized" entertainment product that was more "accommodating the needs of the individual" than AIBO. Later, Sir Stringer lovingly embraced one of Sony's newest products — Sony Reader.

Now, don't get me wrong, there might be some virtue in a portable eBook reader, but doesn't Adobe already do that with Acrobat? And by using a breakthrough hack, you can even install Adobe Acrobat Reader on a Sony PSP™ for transforming this marvelous handheld entertainment system into an, err, eBook reader. (Note: *This hack is thoroughly documented in my upcoming book — Take This Stuff and Hack IT! — due in your bookstore this Fall.*)

It didn't help matters that Dan Brown — author of *The Da Vinci Code* — sheepishly joined Sir Stringer on the stage for begrudgingly discussing the virtues of this new spin on an old idea.

It looked to me like Sir Howard Stringer just doesn't get robots. But he does get gadgets. He even attempted to convince the audience that the Sony Reader would be good for the environment. Yes, that's exactly what our landfills need, more discarded gadgets.

The bottom line is that AIBO was innovative, while the "new" Sony Reader is not.

## If I Only Had a Heart

This techno-love affair with Sony's robotic pooch was more than just circuits and wires deep — AIBO had a personality, a perceived affection by its owners that is tough to find in many of today's other robots. And this type of "true love" is certainly absent from gadget lust.

That same AP news report had several accounts from AIBO owners who claimed that AIBO's artificial intelligence enabled the robot to mirror its behavior to that of its owner. You know the adage about how a dog is a reflection of its master. Well, in this case, the robot was a reflection of its owner.

Able to understand dozens of spoken commands, as well as discriminate the features of three human faces through digital image recognition software, AIBO was as close as you can get to, well, a dog.

Oddly enough, one of the many interesting footnotes in the history of AIBO deals with this robot dog's name. The name — AIBO — can be translated to mean "pal" in Japanese.

So, released in 1999, AIBO enjoyed a mixed life for seven years, when it was cut from Sony's inventory in a cost-savings action. Hardly the way to treat your pal.

One of the most eloquent eulogies of the Sony Corporation Entertainment Robot program was delivered by Mark W. Tilden:

"Personally, I'm sorry that the AIBO and QRIO have faded. The Sony items showed a deep quality that set standards, and their promotion inspired many (including myself). Their killer was their cost, but the idea lives on. Hope to pick up a QRIO from eBay some day. All it needs is a new brain." SV

*Dave Prochnow is a frequent contributor to Nuts & Volts and SERVO Magazine, as well as the author of 26 nonfiction books including the mega-hit The Official Robosapien Hacker's Guide (McGraw-Hill, 2006), the bestselling PSP Hacks, Mods, and Expansions (McGraw-Hill, 2006), and the forthcoming Take This Stuff and Hack IT! (McGraw-Hill, 2007).*

# Then and NOW

## REST IN PEACE, AIBO

### *You Were a Good Dog*

by Tom Carroll

In this column, I generally like to write about a type or style of robot that has been around for decades and has undergone some amazing changes. Quite often, the first of the series of robots that I write about were the "first of their kind," or similar eye-catching appeal. My wife, Sue, commented about one of the headlines about Sony's decision to stop production of their cute little robot dog back in January. As I am writing this in early February, I've been watching some more of those headlines in various magazines and Internet sites: "Sony pulls the plug on its Aibo robot dog," "Sony robot dog rolls over for the last time," "Sony axes robot line," "The dog days of robotics at Sony are coming to an end," and many other


unflattering titles concerning Sony's decision to stop production of entertainment robots.

Aibo lived seven short years, or 49 'dog years' if those count for robot dogs. Seven years is short for a series of robots, but Sony made some spectacular progress in those years. Sony has sold over 150,000 Aibos since launching the entertainment robot line in May 1999. This amazing robot deserves a look back in time to trace its success. Since I am writing to a magazine audience who probably knows quite a bit about this robot, I'd like to touch upon a bit of the history and corporate decisions concerning the product line rather than how it works.

There are so many sensors, cameras, types of software, body styles, and add-ons for Aibo that it would take a shelf of books to cover thoroughly. You, our readers, know that the little fellow has been gutted and hacked more than any robot that we know of, despite its high cost.

Sony's third-quarter fiscal report stated that it would "no longer develop new Aibos. It will continue to support existing models until they're at least each seven years old. That means that the three new Aibos launched in September of 2005 will be supported until March of 2013," said Kirstie Pfeifer, a spokesperson for Sony in San Diego, CA. Sony said that it will continue selling all the Aibos it currently has until it is depleted.

Sony said it plans to shift its research and development in artificial intelligence into future consumer electronics products, but offered no further details. Parts and various consumable items, such as batteries and memories, should be available during this period. Considering the many items still available for old Heath Hero and RB5X robots, it will not be surprising that these robots will have a following for many decades to come.

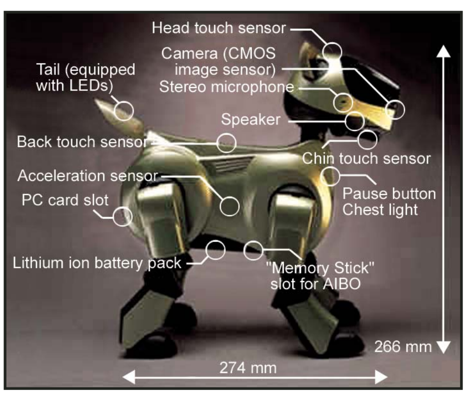
The Tokyo-based consumer electronics giant is restructuring under new chief executive — Sir Howard Stringer. A native of Cardiff, Wales, Sir Howard received the title of Knight Bachelor from Her Majesty Queen Elizabeth II in December of 1999. Married to a medical doctor, Stringer had a long career with CBS TV — from journalist to producer to president.

Stringer is the first non-Japanese head of Sony, a point that many around the world are making about the company's decision to scrap Aibo, the robot dog and Qrio, their humanoid biped robot. He said that Sony plans to focus on three core businesses: electronics, games, and entertainment. It is the profitability game. If their "entertainment" robot dog is not entertainment, then what is it?

Software engineer, Joe Barnhart from Santa Clara, CA, has six Aibos. He says it is because his business travels keep him from caring for the flesh and blood variety. "Of course I'm disappointed in Sony's decision," he said. He feels that the decision to stop entertainment robot development came from Stringer "because he is not Japanese and doesn't understand the importance of robots in Japanese society."

The website for Aibo owners — **www.aibo-life.org** — was filled with comments of disappointment and dismay just after the fiscal report's posting. Bruce Bender, who lives in Rancho Cordova near Sacramento, CA, owns 56 Aibos of all styles and colors which he believes to be the world's largest private collection. That's over $100,000 worth of the robot dogs and lion cubs. Hey, for that amount of money, you could rent Honda's amazing Asimo bipedal robot for the better part of a year.

Bender once mentioned that he still "plans to host gatherings for the worldwide Aibo community," such as one he held in late September of last year where 110 robot dogs danced together in unison. He also noted in a message board posting that although Aibo manufacturing has stopped, "that doesn't mean Aibo is dead."




For the timeline, Sony introduced Aibo in Japan in 1999. This marked a turning point for the world of entertainment. Aibo means "companion" in Japanese and is an acronym for Artificial Intelligence roBOt. When the first 5,000 ERS-110 models were introduced, 3,000 sold on the Internet in Japan within 20 minutes. The 2,000 models that remained were

made available exclusively for the United States and were sold within four days on the Internet at $2,500 each.

People kept asking, "can you make it do more?" Ten thousand of the improved Aibo ERS-111 models, which has a 64-bit microprocessor were made available at the end of 1999 despite receiving 135,000 orders, again at $2,500. An optional 'performer kit' was available for $500. Aibo could heel and chase a ball and later models could do a little dance and wave his front paw on hearing his name.

In later models, speech-recognition software allowed him to learn and respond to dozens of spoken commands and have emotional qualities. Sony claims to have sold 50,000 units of the second-generation ERS-210A — the lion cub look-alike that debuted late 2000.

Most people I have spoken with about the Aibo felt that the 210 still looked like a dog because of the non-cat-like stubby tail. The ears stood up like most dogs — unlike the 'hound dog' floppy ears of the earlier and now, latest models. The LM Series followed in October 2001 and the ERS-220A in November 2001.

Sony pretty much admitted that in the beginning, its popular robot pet did little more than talk back and eat batteries every couple of hours, so they developed software that allowed Aibos to read email messages and web pages. Sony kept up the series of improvements through September of last year when it introduced its final models. The ERS-7M3/T champagne brown Aibo model with the smoothly curved body sells for about $2,100. The pearl white and pearl black models — the ERS-7M3/W and ERS-7M3/B — both sell for about $2,000. The earlier ERS-7M2 Aibo can find its charging station and charge itself as long as it can see the specially marked station tower.

ERS-7M3 includes "short-term" memory that allows Aibo to remember some of the navigation details of a room or floor. So, even if it can't see the charging station, it can still find it. However, Aibo's short-term memory can fade as it travels further and further from its charging station. A wireless card upgrade has allowed 802.11G connectivity to a PC.

The Aibo phenomenon is worldwide and has introduced robotics to many people who would otherwise have overlooked the science. It has always been an expensive "toy" that only a few die-hard owners could afford at $1,500 to $2,500. Because of the costs, there have been several knock-offs — such as the I-Cybie — but none could match the performance and features of even the original ERS-110.

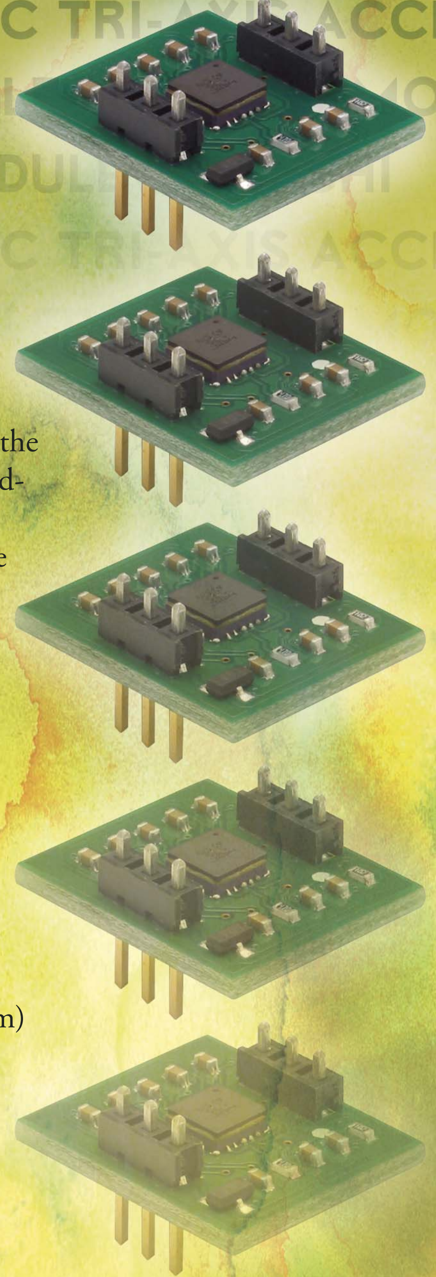
Aibo soccer leagues have been formed, as well as numerous owners groups. Every type of contest imaginable has been held with the little doggies performing all sorts of tricks.

Months later, after the announcement of cessation of production, owners and wanna-be owners still are amazed at Sony's decision.

Aibo may no longer be waddling out Sony's doors into new owners' waiting arms, but his spirit will live long in the hearts of hundreds of thousands of people around the world. SV

# HITACHI H48C TRI-AXIS ACCELEROMETER MODULE

**$39**
**No.28026**



The **Hitachi H48C 3-Axis Accelerometer** is an integrated module that can sense gravitational (g) force of ±3g on three axes (X, Y, and Z). The module contains an onboard regulator to provide 3.3-volt power to the H48C, analog signal conditioning, and an MCP3204 (four channel, 12-bit) analog-to-digital converter to read the H48C voltage outputs. All components are mounted on a breadboard-friendly, 0.7 by 0.8 inch module. Acquiring measurements from the module is simplified through a synchronous serial interface. With the BASIC Stamp® series, for example, this is easily handled with the SHIFTOUT and SHIFTIN commands.

*Features:*

- Measure ±3 g on any axis
- Uses MEMS (Micro Electro-Mechanical System) technology, with compensation for calibration-free use
- Onboard regulator and high-resolution ADC for simple connection to microcontroller host - compatible with BASIC Stamp 2 series SHIFTOUT and SHIFTIN commands
- Free-fall output indicates simultaneous 0g an all axes
- Small, breadboard-friendly package: 0.7” x 0.8” (17.8 cm x 20.3 cm)
- Wide operational range: -25° to 75° C

Order online at **www.parallax.com** 24-hours or call the Parallax Sales Department **toll-free** at **888-512-1024** (Mon-Fri, 7 am-5 pm, PT).

PARALLAX

**WWW.PARALLAX.COM**



**Take part in Parallax workshops at the Maker Faire in San Mateo, CA. April 22-23, 2006**

**Maker Faire**